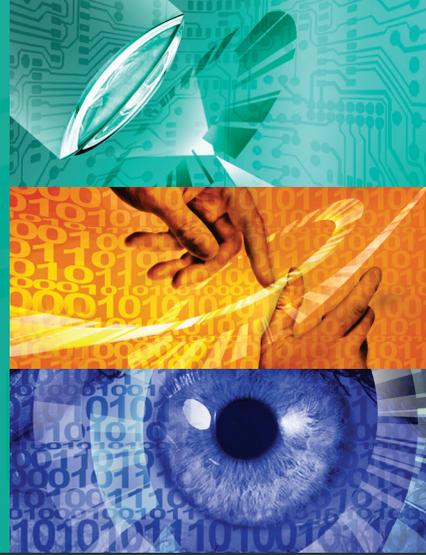


Karlsruher Schriften
zur Anthropomatik

Band 60



Patrick Philipp

**Über die Formalisierung und Analyse
medizinischer Prozesse im Kontext von
Expertenwissen und künstlicher Intelligenz**

Patrick Philipp

**Über die Formalisierung und Analyse
medizinischer Prozesse im Kontext von
Expertenwissen und künstlicher Intelligenz**

Karlsruher Schriften zur Anthropomatik

Band 60

Herausgeber: Prof. Dr.-Ing. habil. Jürgen Beyerer

Eine Übersicht aller bisher in dieser Schriftenreihe
erschienenen Bände finden Sie am Ende des Buchs.

Über die Formalisierung und Analyse medizinischer Prozesse im Kontext von Expertenwissen und künstlicher Intelligenz

von
Patrick Philipp

Karlsruher Institut für Technologie
Institut für Anthropomatik und Robotik

Über die Formalisierung und Analyse medizinischer Prozesse
im Kontext von Expertenwissen und künstlicher Intelligenz

Zur Erlangung des akademischen Grades eines Doktors der Ingenieur-
wissenschaften von der KIT-Fakultät für Informatik des Karlsruher
Instituts für Technologie (KIT) genehmigte Dissertation

von Patrick Philipp

Tag der mündlichen Prüfung: 16. Februar 2023
Erster Gutachter: Prof. Dr.-Ing. habil. Jürgen Beyerer
Zweiter Gutachter: Prof. Dr. Michael Beigl

Impressum



Karlsruher Institut für Technologie (KIT)
KIT Scientific Publishing
Straße am Forum 2
D-76131 Karlsruhe

KIT Scientific Publishing is a registered trademark
of Karlsruhe Institute of Technology.
Reprint using the book cover is not allowed.

www.ksp.kit.edu



*This document – excluding parts marked otherwise, the cover, pictures and graphs –
is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0):
<https://creativecommons.org/licenses/by/4.0/deed.en>*



*The cover page is licensed under a Creative Commons
Attribution-NonCommercial 4.0 International License (CC BY-ND 4.0):
<https://creativecommons.org/licenses/by-nd/4.0/deed.en>*

Print on Demand 2023 – Gedruckt auf FSC-zertifiziertem Papier

ISSN 1863-6489
ISBN 978-3-7315-1289-9
DOI 10.5445/KSP/1000156884

Kurzfassung

Die Digitalisierung hat bereits viele Bereiche der Wirtschaft und des gesellschaftlichen Lebens verändert. Auch unterliegen die Aspekte des Gesundheitswesens und der klinischen Praxis einem digitalen Wandel. Dabei steigt gleichzeitig der Bedarf an weltweit akzeptierbaren Lösungen zum Austausch und zur Auswertung medizinischer Daten. In diesem Zuge gewinnen die Fast Healthcare Interoperability Resources (FHIR) zunehmend an Bedeutung – in Deutschland werden sie beispielsweise in Zusammenhang mit der Elektronischen Patientenakte (ePA) eingesetzt.

Im Hinblick auf diese Entwicklungen, wird FHIR in der vorliegenden Dissertation zur Wissensrepräsentation genutzt. Dabei wird in einem ersten Beitrag die probabilistische Erweiterung eines FHIR DataRequirements vollzogen. Die Erweiterung erlaubt es, Observationswahrscheinlichkeiten zu hinterlegen und so ein Lücke bei der bisherigen Prozessmodellierung in FHIR zu schließen.

Darüber hinaus wird in dieser Arbeit eine strukturierte Vorgehensweise zur Wissensakquisition und -formalisierung von medizinischen Prozessen im Bereich der Diagnostik sowie der Therapie vorgestellt. Das Konzept beschreibt ein Expertensystem, bei welchem FHIR als Wissensbasis eingesetzt wird. Für den Wissenserwerb dienen Ablaufmodelle, die für alle Seiten, sprich: sowohl für den medizinischen als auch für den technischen Domänenexperten, verständlich sind. Dies ist eine wichtige Voraussetzung, um in der tatsächlichen Praxis gemeinsam das Prozesswissen formalisieren zu können.

Grundsätzlich können verschiedene Ablaufmodelle zur Formalisierung herangezogen werden. In dieser Arbeit werden, aufgrund diverser positiver Eigenschaften, UML Aktivitätsdiagramme exemplarisch eingesetzt.

Für das dargestellte Konzept können aber auch andere Modelle, wie z.B. Ereignisgesteuerte Prozessketten (EPK), genutzt werden. Ermöglicht wird dies durch insgesamt drei Transformationsschritte. Sie entkoppeln die Ablaufmodelle von der eigentlichen Wissensbasis. Dabei werden mittels zweier (bereits existierender) Transformationen die Ablaufmodelle zunächst in Workflow-Netze und später in Prozessbäume überführt. Die anschließende Überführung eines Prozessbaums in einen FHIR-Plan ist Beitrag dieser Arbeit. Hierin wird gezeigt, dass die dargelegte Transformation sowohl für Hin- als auch für Rückrichtung strukturerhaltend ist.

Darauf aufbauend wird eine neu entwickelte Transformation eines FHIR-Plans in ein expertenbasiertes Dynamisches Bayes-Netz (DBN) zur Phasenerkennung besprochen. Das Ergebnis wird anhand zweier prototypischer Vertreter chirurgischer Interventionen beleuchtet: die endoprothetische Versorgung eines Patienten mittels einer totalen Endoprothese der Hüfte sowie die laparoskopische Entfernung der Gallenblase. In einer Vor- und Realstudie sowie an einem öffentlichen Datensatz werden die Modelle schließlich einer Leistungsbewertung unterzogen. Zum Einsatz kommt hierfür eine verschachtelte Kreuzvalidierung. Unter Berücksichtigung der gewählten Werte (Precision, Recall, Jaccard-Score) zeigt sich die Überlegenheit des Ansatzes bezüglich der Vergleichsmodelle.

Danksagung

Die vorliegende Dissertation entstand im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter sowohl am Lehrstuhl für Interaktive Echtzeitsysteme (IES) des Karlsruher Instituts für Technologie (KIT), als auch in der Abteilung Interaktive Analyse und Diagnose (IAD) des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildauswertung (IOSB).

An dieser Stelle möchte ich mich bei allen bedanken, die mich während meiner Promotion begleitet und unterstützt haben.

Insbesondere danke ich den Personen, die mir diese Arbeit ermöglicht haben: Meinem Doktorvater und Institutsleiter des Fraunhofer IOSB Professor Dr.-Ing. habil. Jürgen Beyerer für die Förderung meiner Arbeiten, die intensiven Diskussionen in Freitagsrunden und Sommerseminar sowie das stets entgegengebrachte Vertrauen. Ebenso danke ich Herrn Professor Dr. Michael Beigl für die Übernahme des Koreferats und dem Interesse an meiner Arbeit.

Darüber hinaus gebührt mein Dank den Kolleginnen und Kollegen des Lehrstuhls, der Abteilung IAD und insbesondere den Mitarbeitenden meiner Forschungsgruppe Decision Support Systems am Fraunhofer IOSB.

Meiner Frau Lena gilt ein spezieller Dank für ihr entgegengebrachtes Verständnis und genial-denkwürdige Diskussionen.

Inhaltsverzeichnis

Kurzfassung	i
Danksagung	iii
1 Einleitung	1
1.1 Eigene Beiträge	3
1.2 Gliederung	5
2 Wissensbasierte Modellierung	7
2.1 Wissensbasierte Systeme	7
2.2 Expertensysteme	8
2.3 Künstliche Intelligenz	11
2.3.1 Sichten	13
2.3.2 Verbindungen	14
2.4 Bayes-Netze als Komponente zur Problemlösung	16
2.4.1 Wahrscheinlichkeitsbegriffe	16
2.4.2 Wahrscheinlichkeitstheorie	19
2.4.3 Definition	24
2.4.4 d-Separation	28
2.4.5 Dynamische Aspekte	30
2.5 FHIR zur Wissensrepräsentation	34
2.5.1 Motivation	35
2.5.2 Hierarchie der Ressourcen und Elemente	37
2.5.3 Medizinische Entscheidungsunterstützung	39
3 Prozessmodelle	47
3.1 Begrifflichkeiten	47

3.1.1	Prozess	47
3.1.2	Workflow	48
3.1.3	Arbeitsablauf	49
3.1.4	Definitionen	50
3.2	Petri-Netze	52
3.2.1	Notationselemente	52
3.2.2	Verschaltungen	57
3.2.3	Workflow-Netze	60
3.3	UML Aktivitätsdiagramme	63
3.3.1	Notationselemente	63
3.3.2	Verschaltungen	69
3.3.3	FHIR Repräsentationen	69
3.4	Prozessbäume	74
3.4.1	Aufbau und Notationselemente	74
3.4.2	Formale Definition	78
4	Transformationen zwischen Prozessbäumen und FHIR	
	Plänen	81
4.1	Gesamtkonzept	81
4.2	Prozessbaum als Ausgangsmodell	85
4.3	FHIR Plan als Zielmodell	87
4.4	Transformation eines Prozessbaums in einen FHIR-Plan	88
4.5	Transformation eines FHIR-Plans in einen Prozessbaum	92
4.6	Erhaltung der Struktur der Originalmodelle	94
4.6.1	Strukturerhaltung bei Hintransformation	94
4.6.2	Strukturerhaltung bei Rücktransformation	99
5	Assistenz bei chirurgischen Interventionen	101
5.1	Medizinischer Hintergrund	102
5.1.1	Totale Endoprothese der Hüfte	103
5.1.2	Laparoskopische Cholezystektomie	108
5.2	Konstituenten des Problems	113
5.2.1	Begrifflichkeiten	113
5.2.2	Bewertung der Problemlösungen	117
5.3	Stand der Technik	132

5.3.1	Random Forest (RF)	132
5.3.2	Petri-Netz	138
5.3.3	Hidden Markov Model (HMM)	142
5.3.4	Dynamisches Bayes-Netz (DBN)	144
5.3.5	Zusammenfassung	146
5.4	Transformation eines FHIR-Plans in ein DBN zum Abgleich chirurgischer Interventionen	150
5.4.1	FHIR-Plan als Ausgangsmodell	150
5.4.2	DBN als Zielmodell	152
5.4.3	Transformation eines FHIR-Plans in ein DBN	154
5.5	Vorstudie	163
5.5.1	Material und Methoden	163
5.5.2	Ergebnisse	166
5.6	Realstudie	168
5.6.1	Material und Methoden	169
5.6.2	Ergebnisse	178
5.7	Galle	186
5.7.1	Material und Methoden	186
5.7.2	Ergebnisse	189
6	Schlussbetrachtung	197
6.1	Fazit	198
6.2	Ausblick	199
	Abkürzungen	203
	Medizinischer Glossar	207
	Symbolverzeichnis	209
	Literatur	215
	Eigene Publikationen	237
	Betreute studentische Arbeiten	245

1 Einleitung

Künstliche Intelligenz (KI) wird viele Aspekte der klinischen Praxis und des Gesundheitswesens entscheidend mitprägen. Dies betrifft nach [Yu18] u.a.:

- die Unterstützung bei der Diagnosestellung,
- die Assistenz bei chirurgischen Interventionen,
- die Auswertung der Ergebnisse bildgebender Verfahren,
- die Unterstützung in der Arzneimittelforschung.

Die Beiträge der vorliegenden Dissertation zur Formalisierung und Analyse von medizinischen Prozessen lassen sich in den beiden erstgenannten Bereichen verorten (vgl. Abbildung 1.1). Sprich: der Bestimmung der zugrundeliegenden Erkrankung eines Patienten und der Assistenz bei chirurgischen Behandlungsmaßnahmen.

Als Grundlage für Künstliche Intelligenz (KI)-Methoden stehen klassischerweise große Datenmengen im Fokus. Im Bereich der bildgebenden Verfahren können beispielsweise anhand großer Lern-, Validierungs- und Teststichproben Tumore in Bildschichten segmentiert oder Anomalien erkannt werden.

In anderen Bereichen stehen geeignete Daten nur eingeschränkt oder unstrukturiert zur Verfügung. Zum Beispiel liegen bei Diagnosestellung häufig große Teile der Daten in Form von Freitexten vor und sind damit für datenbasierte KI-Methoden nicht oder nur unzureichend nutzbar. Um die Datenschätze mittels Methoden der KI zu bergen, wird deswegen üblicherweise eine Strukturierung der Daten angestrebt. In der Vergangenheit entstand so ein ganzer Strauß an verschiedenen Insellösungen zur Strukturierung und Speicherung. Gleichzeitig erschweren Insellösungen aber Daten auf weltweiter

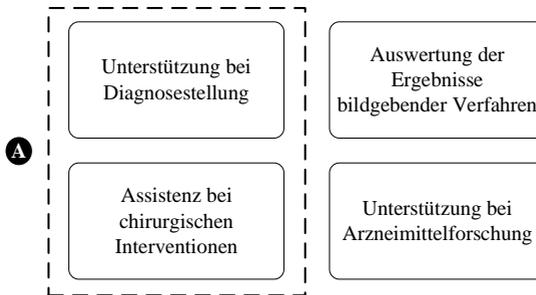


Abbildung 1.1: Wichtige Aspekte auf dem Gebiet des Gesundheitswesens und der klinischen Praxis, in denen künstliche Intelligenz nach [Yu18] einen prägenden Einfluss haben wird. Die Beiträge der vorliegenden Dissertation liegen in den markierten Bereichen d. h. bei der Unterstützung der Diagnosestellung sowie der Assistenz bei chirurgischen Interventionen (vgl. Marker A).

Ebene zu sammeln, auszutauschen und für nachfolgende Auswertungen zu nutzen. Abseits von Insellösungen entstand somit der Bedarf an weltweit akzeptierbaren Lösungen und Interoperabilität.

In diesem Zuge gewannen die sogenannten Fast Healthcare Interoperability Resources (FHIR) in den letzten Jahren zunehmend an Bedeutung [Leh19a, Hub20]. Schon im Jahr 2018 verständigten sich die Technologieunternehmen IBM, Microsoft, Oracle, Google, Amazon und Salesforce in einer gemeinsamen Erklärung auf die Notwendigkeit, bestehende Hemmnisse bei der medizinischen Datenrepräsentation zu beseitigen. In dieser Erklärung werden FHIR und angeschlossene Projekte bereits explizit genannt [Inf18, Leh19a], denn sie sollen es ermöglichen, medizinische Daten mit Methoden der KI weltweit, sowie unternehmensübergreifend effektiv und effizient nutzbar zu machen [Leh19b].

Weiterhin ergab sich speziell in Deutschland die Frage nach akzeptierbaren Softwarelösungen im Hinblick auf die Elektronische Patientenakte (ePA), deren Entwicklung durch entsprechende Gesetze in den Jahren zuvor angeregt wurden. Im Jahr 2019 stellten hierzu [Plo19] ein Konzept vor, welches FHIR

nutzt, um durch eine möglichst vertrauenswürdige Lösung eine hohe Akzeptanz bei den Benutzern zu erreichen. Heute, im Jahr 2022, basieren Schnittstellen, sowie die auf der ePA ablegbaren und in Bearbeitung befindlichen medizinischen Informationsobjekte (MIO), wie beispielsweise Impfpass, Mutterpass oder das Zahnärztliche Bonusheft, auf FHIR [Kas21].

Abgeleitet aus den genannten Entwicklungen, nutzt die vorliegende Dissertation FHIR als Vehikel zur Wissensrepräsentation im Rahmen der Prozessmodellierungen und erweitert dieses wo notwendig.

Besonders in Bereichen, die datenarm aber wissensreich sind, kann darüber hinaus das verfügbare Expertenwissen eine entscheidende Rolle spielen. Dies gilt beispielsweise für medizinische Prozesse im Rahmen der Diagnosestellung oder der chirurgischen Interventionen, die interessanterweise häufig noch unzureichend unter den oben genannten Gesichtspunkten betrachtet werden. Wichtig werden solcherlei Überlegungen vor allem dann, wenn am Ende eine praxistaugliche Assistenz im Rahmen medizinischer Prozesse konzipiert und bereitgestellt werden soll. Aus diesem Grunde beschäftigen sich die Beiträge der vorliegenden Arbeit auch mit diesem Aspekt.

1.1 Eigene Beiträge

Einer der Beiträge dieser Dissertation ist eine strukturierte Vorgehensweise zur Wissensakquisition und -formalisierung von medizinischen Prozessen für eine KI-basierte Entscheidungsunterstützung im Bereich der medizinischen Diagnostik und Therapie, vgl. hierzu Abbildung 1.2.

Medizinische Prozesse werden dabei durch Prozessmodelle beschrieben. Hierzu muss das vorhandene Expertenwissen im Dialog mit dem Experten formalisiert werden. Ziel der Formalisierung ist letztlich ein für alle Seiten verständliches Ablaufmodell, welches den Ablauf der einzelnen Prozessschritte des medizinischen Prozesses beschreibt. Grundsätzlich können für ein solches Ablaufmodell unterschiedliche Prozessmodelle herangezogen werden. Aufgrund ihrer Verständlichkeit auch für nicht-technische Nutzer eignen sich

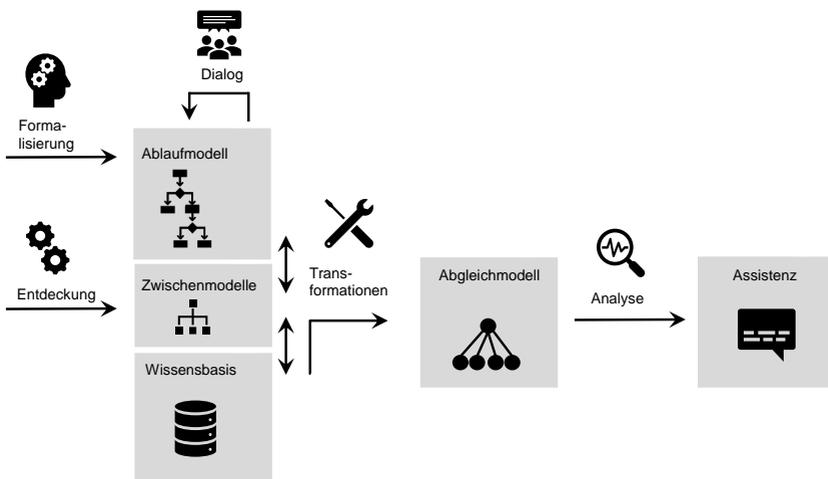


Abbildung 1.2: Die Formalisierung des menschlichen Wissens kann im Dialog der medizinischen und technischen Domänenexperten vollzogen werden. Ergebnis ist ein für alle Seiten verständliches Ablaufmodell, welches über Zwischenmodelle in die FHIR-basierte Wissensbasis gelangt. Zusätzlich können Methoden zur Entdeckung von Abläufen (vgl. Process Mining) eingebunden werden. Aus der Wissensbasis können wiederum Abgleichmodelle für die Bereitstellung von Assistenzfunktionen erzeugt werden.

insbesondere grafische Prozessmodelle, wie beispielsweise Geschäftsprozessmodell und -notation (BPMN), Ereignisgesteuerte Prozessketten (EPK) oder UML Aktivitätsdiagramme [Ko09]. In dieser Arbeit werden UML Aktivitätsdiagramme zur Modellierung und den Wissenserwerb genutzt. Die Beiträge der Arbeit sind allerdings von den eben genannten, konkreten Prozessmodellen über Zwischenmodelle entkoppelt, sodass diese mit alternativen Modellen zusammenspielen können. Weiterhin bieten die Zwischenmodelle eine Nahtstelle zu unterschiedlichen Techniken für das datenbasierte Entdecken von Prozessen.

Die in den Prozessmodellen enthaltene Information muss schließlich für eine Wissensbasis verfügbar gemacht werden. Dafür werden die im Expertendialog erstellten UML Aktivitätsdiagramme in FHIR überführt, sodass die Abläufe schließlich als FHIR-Pläne hinterlegt werden können. Dies erfolgt durch

drei Transformationen, von denen zwei bereits Stand der Technik sind. Die dritte Transformation stellt einen eigenen Beitrag dar – hierin werden Prozessbäume und FHIR-Pläne hin- und rücktransformiert.

In einem weiteren Beitrag wird eine probabilistische Erweiterung einer FHIR Ressource vorgestellt und schließlich wird in einem finalen Beitrag ein probabilistisches Abgleichmodell für die Assistenz im Rahmen chirurgischer Interventionen abgeleitet. Das Abgleichmodell wird dann anhand zweier prototypischer Vertreter evaluiert.

1.2 Gliederung

Die vorliegende Arbeit enthält sechs Kapitel, deren Inhalt nachstehend skizziert wird:

Kapitel 2 beleuchtet die Grundlagen der wissensbasierten Modellierung. Hierbei wird zunächst auf Expertensysteme und deren Zusammenhang mit Künstlicher Intelligenz eingegangen. Weiterhin werden Bayes-Netze als Problemlösungskomponente ausführlich erläutert und schließlich die sogenannten Fast Healthcare Interoperability Resources zur Wissensrepräsentation dargestellt sowie erweitert.

Kapitel 3 widmet sich den Begrifflichkeiten der Prozessmodellierung und stellt Notationselemente sowie Verschaltungen von Petri-Netzen, UML Aktivitätsdiagrammen als auch Prozessbäumen vor.

Kapitel 4 stellt die entwickelte Transformation zwischen Prozessbäumen und FHIR dar. Zusätzlich wird die Strukturhaltung bei Hin- und Rücktransformation beleuchtet.

Kapitel 5 beschreibt die medizinischen Hintergründe zweier prototypischer chirurgischer Intervention: Zum einen die laparoskopische Cholezystektomie und zum anderen die totale Endoprothese der Hüfte. Darauf aufbauen wird auf die Konstituenten des zugrunde liegenden Problems und die Bewertung der Problemlösung eingegangen.

Schließlich wird ausgehend vom Stande der Technik der eigene Beitrag umrissen und schließlich detailliert vorgestellt.

Kapitel 6 fasst die Erkenntnisse zusammen und gibt einen Ausblick.

Diese Arbeit enthält neben den beschriebenen Kapiteln einen medizinischen Glossar (ab Seite 207), ein Symbolverzeichnis (ab Seite 209), ein Literaturverzeichnis (ab Seite 215), ein Verzeichnis der eigenen Publikationen (ab Seite 237) und ein Verzeichnis betreuter studentischer Arbeiten (ab Seite 245).

2 Wissensbasierte Modellierung

Wissensbasierte Modellierung steht in engem Zusammenhang mit dem Forschungsgegenstand der Wissensbasierte Systeme (WBS), die in Kapitel 2.1 vorgestellt werden. Speziell auf den Aufbau von Expertensystemen wird im anschließenden Kapitel 2.2 eingegangen. WBS und Expertensysteme haben sich als Teilgebiet der KI entwickelt – ein Überblick und eine Einordnung wird in Kapitel 2.3 vorgenommen. Auf Wahrscheinlichkeitsbegriffe und -theorie wird in Kapitel 2.4 gemeinsam mit Bayes-Netze als Problemlösungskomponente eingegangen. Schließlich werden als Grundlage zur Wissensrepräsentation die sogenannten Fast Healthcare Interoperability Resources (FHIR) in Kapitel 2.5 vorgestellt.

2.1 Wissensbasierte Systeme

Über die Zeit hinweg finden sich verschiedene Definitionen des Begriffs WBS [Ort77, Kur92, Ake09, Pup13b]. Allen gemein ist die Aufspaltung der Architektur des Systems in zwei Bereiche (vgl. Abbildung 2.1). Auf der einen Seite steht die sogenannte Wissensbasis mit spezifischem Wissen über den Anwendungsbereich. Dem gegenüber steht die tatsächliche Verarbeitung des Wissens (Wissensverarbeitung) in Form einer anwendungsunabhängigen Problemlösungskomponente.

An Anlehnung an die Definitionen aus [Ort77, Kur92, Ake09, Pup13b], wird in der vorliegenden Arbeit ein WBS definiert als:

Definition 2.1. Ein WBS ist ein System oder Programm, bei dem das Wissen über ein Anwendungsgebiet (Wissensbasis) getrennt ist von allgemeinen Problemlösungsmethoden (Wissensverarbeitung).

Diese Trennung hat zwei Vorteile: Zum einen können allgemeine Methoden zur Wissensverarbeitung herangezogen werden, die unabhängig von der Wissensbasis eines konkreten Anwendungsfalles sind und die auch auf weitere Anwendungen übertragbar sind. Zum anderen kann die Wissensbasis aktualisiert und erweitert werden, ohne dass dabei die Algorithmen der Wissensverarbeitung angepasst werden müssen [Kol09].

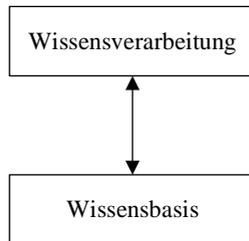


Abbildung 2.1: Auch wenn es verschiedenartige Definitionen des Begriffs WBS gibt, ist die Trennung zwischen Fachwissen über das Anwendungsgebiet des WBS und der eigentlichen Wissensverarbeitung mittels allgemeiner Problemlösungsmethoden allen Definitionen gemein.

2.2 Expertensysteme

In der Literatur findet sich üblicherweise eine Abgrenzung des Begriffs des Expertensystems zum Begriff WBS. Hierbei existieren verschiedene Sichtweisen, von einigen Autoren werden die Begriffe auch synonym verwendet [Hop92]. Eine gängige Möglichkeit ein Expertensystem von einem WBS abzugrenzen ist die Herkunft des im System repräsentierten Wissens. Dabei sind Expertensysteme spezielle WBS, deren Wissen von Experten stammt [Wat86].

Sie können somit als Teilmenge der WBS angesehen werden. Bei dieser Definition bleibt es Auslegungssache, ob die Wissensbasis eines Expertensystems ausschließlich aus Expertenwissen bestehen muss. Denkbar ist auch eine datengetriebene Erweiterung der Wissensbasis durch Methoden des maschinellen Lernens.

In der vorliegenden Arbeit wird für Expertensysteme eine ähnliche Definition nach Puppe herangezogen [Pup13b]:

Definition 2.2. Ein Expertensystem ist ein WBS, mit dem die Schlussfolgerungsfähigkeit und das bereichsspezifische Wissen von Experten nachgebildet werden soll.

Diese Definition umfasst damit zwei Aspekte:

1. Expertensysteme sind Teilmenge der WBS.
2. Der Begriff Expertensystem entstammt einer externen Sichtweise und beschreibt, wie die Umwelt das System wahrnimmt. Speziell bleibt dabei frei, wie das nachzubildende bereichsspezifische Wissen erworben wird. Denkbar sind verschiedene Wissensquellen (beispielsweise Datenbanken, Bücher, Leitlinien oder menschliche Experten [Bei19]), sowie die (teilweise) Generierung der Wissensbasis mit dem Werkzeug des Maschinellen Lernens (beispielsweise Erlernen von Entscheidungsbäumen zur Entwicklung der Wissensbasis [Bei19]).

Die Architektur eines Expertensystems besteht aus einzelnen Komponenten, die in einem Austausch miteinander stehen. Grundlegend sind aber nur die Wissensbasis und die Problemlösungskomponente.

Die Wissensbasis bildet dabei den Kern eines Expertensystems – in ihr wird das benötigte Fachwissen über das Anwendungsgebiet des Expertensystems in einer geeigneten Repräsentationsform abgelegt [Kur92]. Unabhängig von der Art der Wissensrepräsentation kann der Inhalt einer Wissensbasis nach

[Pup13a] weiter unterteilt werden in sogenanntes fallspezifisches Wissen und bereichsbezogenes Expertenwissen¹.

Dabei bezieht sich das fallspezifische Wissen ausschließlich auf eine konkrete, gerade betrachtete Problemstellung im Anwendungsgebiet des Expertensystems. Fallspezifisches Wissen kann aufgrund von Beobachtungen – im medizinischen Bereich beispielsweise aufgrund von Untersuchungsergebnissen eines Patienten – vorliegen. Das fallspezifische Wissen kann vom Benutzer des Expertensystems während der Nutzung eingegeben [Kna93] oder ohne direkte Benutzerbeteiligung erhoben werden [Bie13].

Das bereichsbezogene Expertenwissen bezieht sich im Gegensatz zum fallspezifischen Wissen auf Wissen, das über einen einzelnen konkreten Fall aus dem Anwendungsgebiet des Expertensystems hinaus geht – es meint damit Wissen, das alle Fälle aus dem Anwendungsgebiet umfasst. Es kann sich dabei um Fachwissen der Domänenexperten handeln – wie beispielsweise das Wissen über den Ablauf einer Operation.

Am Anfang der Entwicklung eines Expertensystems steht die Akquisition des Fachwissens aus der Anwendungsdomäne. Zur Akquisition des Wissens dient im Expertensystem eine sogenannte Wissenserwerbskomponente², die den Aufbau der Wissensbasis unterstützt [Bei19, Pup13b].

Da unterschiedliche Quellen für den Wissenserwerb in Frage kommen [Bei19], z.B. Leitlinien oder menschliche Experten, muss das Expertenwissen aus diesen Quellen im Schritt der Wissensakquisition geeignet erhoben und in einem Wissensmodell dargestellt werden können [Kur92].

Eine nicht zu unterschätzende Herausforderung ist dabei die Akquisition von Wissen mittels Expertendialog [Alt13]: Da das domänenspezifische Wissen eines Experten häufig nicht bewusst vorhanden ist, sondern unbewusst angewandt wird, gestaltet sich dessen Ermittlung und Strukturierung in der Praxis

¹ Hinweis: Beierle unterteilt ähnlich in fallspezifisches Wissen und regelhaftes Wissen [Bei19]. Wobei letzteres im Sinne eines allgemeinen bzw. bereichsbezogenen Wissens zu interpretieren ist und nicht im Sinne der Form der Repräsentation in Form von Regeln.

² Hinweis: Einige Autoren sprechen hier auch von der Wissensakquisitionskomponente (vgl. [Kur92, Alt13]).

häufig schwierig. Gleichzeitig kann das Ergebnis der Wissensakquisition aber maßgeblich für den Erfolg des Expertensystems verantwortlich sein. Der Prozess des Wissenserwerbs sollte daher sorgfältig vorbereitet und durchgeführt werden [Alt13].

Deswegen ist einer der Beiträge dieser Arbeit eine strukturierte Vorgehensweise für die Wissensakquisition und -formalisierung für Expertensysteme im Bereich der medizinischen Diagnostik und Therapie.

2.3 Künstliche Intelligenz

Als Ursprung des Forschungsfeldes KI wird üblicherweise eine Arbeitstagung zur Maschinenintelligenz angesehen, die im Jahre 1956 von John McCarthy am Dartmouth College in New Hampshire organisiert wurde [McC04, Rus09]. Dabei erklärte McCarthy sinngemäß: „Ziel der KI ist es, Maschinen zu entwickeln, die sich verhalten, als verfügten sie über Intelligenz“¹.

Als eine Möglichkeit eine solche Maschinenintelligenz zu messen, galt zunächst der Turing Test. Dabei tritt ein menschlicher Fragesteller mit zwei verschiedenen Gesprächspartnern in Kontakt, zu denen er weder Sicht- noch Hörkontakt hat.

Über eine Tastatur und einen Bildschirm kann er mit den Gesprächspartnern kommunizieren. Kann der Fragesteller auch nach intensiver Befragung nicht sagen, welcher der beiden Gesprächspartner die Maschine ist, so hat die Maschine den Turing Test bestanden. Der Maschine wird dann ein dem Menschen ebenbürtiges Denkvermögen unterstellt [Tur50]².

¹ Hinweis: Schon zuvor hatte Alan Turing die zentrale Frage „Können Maschinen denken?“ formuliert [Tur09]. Eine Frage die Hermann Schmidt sowie Norbert Wiener (vgl. das Wiener Filter bzw. Wiener-Kolmogoroff-Filter) mit dem Begriff „Kybernetik“ verknüpften [Wie48]. Nachhaltig in den 1960er Jahren geprägt von Karl Steinbuch [Ste65], ist die Kybernetik heute weitestgehend im Begriff der KI aufgegangen.

² Hinweis: Die Testergebnisse hängen also nicht von der Fähigkeit der Maschine ab, „korrekte“ Antworten im Hinblick auf die Konstituenten des Problems zu geben, sondern nur davon, wie sehr ihre Antworten denen eines Menschen (bzw. der Menschen der Stichprobe) ähneln.

Jedoch wurde, insbesondere von dem Philosophen John Searle, Kritik am Turing Test hervorgebracht: er prüfe nur auf Funktionalität und nicht auf Verständnis. Um dies darzustellen, zog Searle das Gedankenexperiment des „Chinesischen Zimmers“ heran, vgl. [Sea80]. Dieses Gedankenexperiment beginnt zunächst mit der Hypothese, dass es gelungen ist eine Maschine zu bauen, die sich so verhält, als verstünde sie Chinesisch, d.h. die Maschine kann basierend auf einem chinesischen Eingabetext einen passenden Ausgabetext so bereitstellen, dass sie im Turing Test einen chinesischen Muttersprachler überzeugen kann.

Laut Searle stellt sich nun die Frage, ob eine solche Maschine tatsächlich Chinesisch versteht, oder ob sie das Verständnis des Chinesischen nur simuliert. Um diesen Unterschied zu verdeutlichen, beschreibt Searle eine angepasste Version des Turing Tests: Auch in seinem Experiment kommuniziert ein Fragesteller ohne Sicht- und Hörkontakt mit einem Gesprächspartner, der sich in einem anderen Zimmer befindet. In diesem Fall ist der Fragesteller jedoch chinesischer Muttersprachler und stellt seine Fragen demnach in chinesischen Schriftzeichen. Sein Gegenüber, in diesem Fall ein Mensch, ist des Chinesischen nicht mächtig. Hat er im Zimmer aber Zugriff auf eine Schriftsammlung in chinesischen Zeichen, die viele verschiedene Fragen und dazu passende Antworten enthält, kann er die Sequenzen an Schriftzeichen in den Fragen des Fragestellers auf der Ebene der Zeichenerkennung mit den Fragen in der Schriftsammlung vergleichen und so die passenden Antwortzeichen herausfinden. Hierbei muss er weder Frage noch Antwort verstehen, kann aber seinem Gegenüber vorgaukeln, er verstünde Chinesisch.

Kann also analog eine Maschine auf in chinesischer Schrift gestellte Fragen mit passenden chinesischen Schriftzeichen antworten, sodass sie den Turing Test besteht, kann hierdurch noch nicht angenommen werden, dass die Maschine tatsächlich versteht was sie übersetzt – also eine Art von Bewusstsein für die Problemstellung hat [Sea80].

2.3.1 Sichten

Ausgehend von diesen frühen Versuchen den Begriff der KI zu fassen und zu vermessen, haben sich über die Zeit eine Vielzahl an Definitionen entwickelt, die teils kontrovers diskutiert werden [Mon20]. In Folge findet sich in der Literatur auch heute keine einheitliche Definition der KI (engl. Artificial Intelligence (AI)). Vielmehr wird der Begriff aus unterschiedlichen Sichten heraus nutzbar gemacht.

Eine dieser Sichten betrifft das Vorhandenseins eines Bewusstseins seitens der Maschine (vgl. Tabelle 2.1). In diesem Sinne lässt sich basierend auf Searle die KI in „stark“ (engl. strong) bzw. „schwach“ (engl. weak) einteilen [Sea80]. Man spricht von starker KI, sofern eine Maschine zu Selbsterkenntnis und Bewusstsein fähig ist und autonom handelt – die Maschine weiß in diesem Sinne also, was sie tut, warum sie es tut, und ist in der Lage, selbständig neue Aufgaben zu finden und diese zu reflektieren. Im Gegenzug spricht man von schwacher KI, wenn eine Maschine Verständnis nur „vortäuscht“, tatsächlich jedoch unreflektiert und ohne Bewusstsein eine Aufgabe abarbeitet [Sea80].

Eine weitere Sicht nimmt Bezug auf die Art der Aufgabe, die mittels KI bewältigt werden kann, indem zwischen einer sogenannten „eingeschränkten“ (engl. narrow) und „allgemeinen“ (engl. general) KI unterschieden wird¹. Die bislang erzielten großen Erfolge im Forschungsfeld der KI sind dem Bereich eingeschränkten KI zuzuschreiben: Zwar können Maschinen Menschen übertreffen, jedoch nur in einer eng gefassten Aufgabenstellung (beispielsweise Go-Spiel, Maschinelles Übersetzen) [Bun17, Kur05]. Im Falle der allgemeinen KI, ist eine Maschine unabhängig von einem konkreten Anwendungsfall in der Lage, verschiedenartige Probleme aus unterschiedlichen Bereichen anzugehen [Mon20].

Wie eine Problemstellung innerhalb der KI repräsentiert und eingebracht wird, wurde schon zu Beginn der KI-Forschung intensiv untersucht. Die

¹ Hinweis: Das englische Wort „narrow“ wird in der deutschsprachigen Literatur teils mit „schwach“ übersetzt. Unabhängig davon wird das englische Wort „general“ auch mit „universell“ übersetzt.

sogenannte „symbolische“ (engl. symbolic) KI stellt eine Sichtweise dar, bei der eine symbolische Ebene [New76, New07], beispielsweise in Form von durch den Menschen vorgegebener Regeln, genutzt wird¹. Aus der heutigen Perspektive wird dieser Ansatz teilweise auch als „Top-Down“ Ansatz bezeichnet, während abgrenzend dazu der „Bottom-Up“ Ansatz in der Anwendung subsymbolischer Methoden des Maschinellen Lernens im Sinne der Problemstellung besteht.

Letzterer wird, gerade bei Verwendung von Künstlichen Neuronalen Netzen, mit dem Begriff der „konnektionistischen“ (engl. connectionist) oder „neuro-nalen“ (engl. neural) KI verknüpft [Smo87, Smo90].

Tabelle 2.1: Drei Sichten auf den Begriff der KI. Die Sichten beleuchten verschiedene Aspekte, wie das Vorhandensein eines Bewusstseins, der Universalität der Intelligenz oder der Repräsentation der Problemstellung.

Sicht auf KI	Ausprägungen
Bewusstsein	Schwache oder starke KI (engl. weak or strong AI)
Universalität	Eingeschränkte oder allgemeine KI (engl. narrow or general AI)
Repräsentation	Symbolische oder subsymbolische KI (engl. symbolic or subsymbolic AI)

2.3.2 Verbindungen

Das Forschungsfeld der KI ist mit verschiedenen weiteren Forschungsfeldern verknüpft. Im Bereich der Informatik bestehen starke Verbindungen zu den

¹ Hinweis: In diesem Zusammenhang wird die symbolische KI auch als Good Old-Fashioned Artificial Intelligence (GOFAI) bezeichnet [Hau89] und bezieht sich damit auf die symbolische Repräsentation der Wissensbasen früherer Expertensysteme.

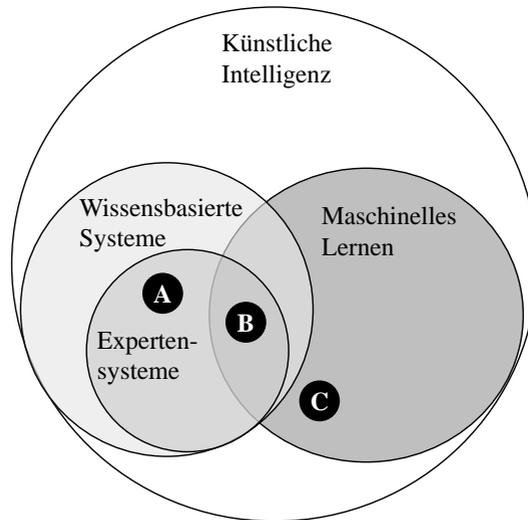


Abbildung 2.2: Die Abbildung zeigt ein Mengendiagramm zur Verdeutlichung der Zusammenhänge zwischen den Begriffen Künstliche Intelligenz, Maschinelles Lernen, Wissensbasierte Systeme und Expertensysteme. Die Punkte A, B und C verdeutlichen den Schnitt dieser Mengen mit der Menge der Künstlichen Intelligenz. Dabei nutzt KI die Methoden des Maschinellen Lernens (C). Neben dem Aspekt des Lernens spielt auch die Wissensbasierte Modellierung in Form von Wissensbasierten Systemen und insbesondere Expertensystemen eine Rolle. Letztere können rein auf Formalisierung von Expertenwissen (A) oder auf einer Kombination mit erlerntem Wissen (B) basieren.

Methoden des Maschinellen Lernens. Hierbei kann KI die Methoden des Maschinellen Lernens auf symbolischer Ebene (bspw. über Entscheidungsbäume) oder subsymbolischer Ebene (bspw. Neuronale Netze) nutzen (vgl. Abbildung 2.2, Punkt C). Die Teilmenge des Bereichs des Maschinellen Lernens, die sich speziell mit mehrschichtigen Neuronalen Netzen beschäftigt, wird auch Tiefes Lernen (engl. Deep Learning [Goo16]) genannt – diese Teilmenge ist aus Gründen der Übersichtlichkeit nicht Bestandteil der Abbildung.

Die Begriffe dieser Mengen werden teils synonym mit dem Begriff der KI verwendet. Sie werden hier aber als mögliche Werkzeuge zur Umsetzung von KI gesehen.

Für das Lösen konkreter Anwendungsprobleme im Rahmen der KI sind neben dem Aspekt des Lernens jedoch auch weitergehende Aspekte von Interesse. Beispielsweise die wissensbasierte Modellierung im Kontext der Expertensysteme, die u.a. die Trennung von Wissensbasis und Wissensverarbeitung vorsieht (vgl. Kapitel 2.1 und 2.2 ab Seite 7). Dabei kann die Wissensbasis klassisch durch Formalisierung menschlichen Vorwissens befüllt werden (A). Eine Kombination mit Methoden des Maschinellen Lernens, also der Kombination von menschlichen Vorwissen und Wissen aus Daten, ist dabei ein vielversprechender aber auch herausfordernder Ansatz der im Rahmen dieser Arbeit verfolgt wird (B)¹.

2.4 Bayes-Netze als Komponente zur Problemlösung

Zur Einführung werden verschiedene Wahrscheinlichkeitsbegriffe in Kapitel 2.4.1 beleuchtet. Daran anschließend folgen in Kapitel 2.4.2 die Grundlagen der Wahrscheinlichkeitstheorie. In den darauf folgenden Kapiteln 2.4.3 und 2.4.4 werden Bayes-Netze genauer besprochen.

2.4.1 Wahrscheinlichkeitsbegriffe

Historisch haben sich unterschiedliche Deutungen des Wahrscheinlichkeitsbegriffs herausgebildet – die frequentistische Sichtweise und die Bayes'sche Sichtweise – auf die im Folgenden kurz eingegangen wird.

¹ Hinweis: Teils wird die Verbindung von KI mit menschlichem (Vor-)wissen auch als Hybride Intelligenz bezeichnet [Kam16, Del19].

Die frequentistische Sichtweise wurde erstmals im 19. Jahrhundert von John Venn als mathematische Theorie ausgearbeitet, vgl. [Ven66]. Diese Sichtweise interpretiert die Wahrscheinlichkeit für das Eintreten eines Ereignisses als die relative Häufigkeit, mit der dieses Ereignis in einer unendlichen Anzahl gleicher und voneinander unabhängiger Zufallsexperimente auftritt. Wahrscheinlichkeit wird in der frequentistischen Sicht also auf einen bestimmten physikalischen Prozess bzw. dessen Ergebnis bezogen.

Damit kann der frequentistische Wahrscheinlichkeitsbegriff dann angewendet werden, wenn Ereignisse beliebig oft und unter gleichen Umständen wiederholbar sind oder dies zumindest denkbar wäre. So beispielsweise beim Münzwurf oder beim Erfassen einer Messgröße. Nicht möglich ist jedoch eine Bewertung von einmaligen Ereignissen („Wie hoch ist die Wahrscheinlichkeit, dass es morgen Nachmittag regnet?“) oder von Hypothesen („Es gibt Leben in anderen Galaxien“).

Dies gelingt jedoch mittels des Bayes'schen Wahrscheinlichkeitsbegriffes, der den frequentistischen Wahrscheinlichkeitsbegriff erweitert [Kor10] und es insbesondere erlaubt, Vorwissen, beispielsweise von Experten, einzubeziehen [Mor90].

Die Bayes'sche Deutung interpretiert Wahrscheinlichkeit als ein Maß dafür, wie sehr man – unter Berücksichtigung vorhandenen Wissens, Annahmen und Daten – vom Eintreten eines Ereignisses überzeugt ist [Leó19]. Man spricht daher auch von der Interpretation der Wahrscheinlichkeit als „Degree-of-Belief“ (deutsch: Grad-des-Dafürhaltens), der einen Wert zwischen 0 (unglaublich, falsch) und 1 (glaubwürdig, wahr) annehmen kann. Wahrscheinlichkeit ist damit also nicht nur eine Funktion des Ereignisses an sich, sondern zusätzlich abhängig vom Wissensstand [Mor90].

Innerhalb der Bayes'schen Deutung gibt es noch einmal zwei Strömungen mit unterschiedlichen Interpretationen des Grad-des-Dafürhaltens: Die subjektive Deutung und die objektive Deutung [Pre09].

Erstere interpretiert den Bayes'schen Wahrscheinlichkeitsbegriff als persönliche Überzeugung eines Individuums. Der Grad-des-Dafürhaltens wird also als

eine individuelle, subjektive Meinung aufgefasst, die so gewählt werden sollte, dass sie konform zu den Axiomen nach Kolmogorov [Kol33] ist [Mor90]. Beispielsweise sollten sich der Grad-des-Dafürhaltens, dass ein Ereignis passiert und der Grad-des-Dafürhaltens, dass dieses nicht passiert, zu 1 addieren. Beim Erheben von Wahrscheinlichkeiten werden daher spezielle Techniken angewandt [Win67], sodass sich anschließend eine zu den Axiomen nach Kolmogorov konforme Wahrscheinlichkeitsverteilung ableiten lässt [Leo54].

Die objektive Deutung sieht im Grad-des-Dafürhaltens im Gegensatz zu dessen subjektiver Deutung keine individuelle Meinung. Der Grad-des-Dafürhaltens steht hier für eine allgemeine, neutrale Erwartung, die jeder mit dem gleichen Wissensstand teilen sollte. So sollte beispielsweise bei Nichtvorhandensein relevanter Evidenzen der Grad-des-Dafürhaltens gemäß des Indifferenz-Prinzips¹, gleichmäßig auf alle in Betracht kommenden möglichen Ereignisse verteilt werden [Eva19].

Auf dem Konzept der Maximalen Entropie basieren zudem weitere Methoden zur Konstruktion allgemeiner A-priori-Wahrscheinlichkeitsverteilungen [Jay88]. Jeffrey schlug spezielle, sogenannte „nicht-informative“ Referenz-Wahrscheinlichkeitsverteilungen vor, die gegenüber einer Reparametrisierung der Modellparameter invariant sind [Jef46].

Für diese Arbeit von Bedeutung ist die subjektive Interpretation des Grad-des-Dafürhaltens, da sie es erlaubt, vorhandenes Expertenwissen durch Expertenbefragungen gezielt in die Wissensbasis einfließen zu lassen. Eine Herausforderung dabei ist, dass der Grad-des-Dafürhaltens in diesem Fall abhängig vom spezifischen Wissensstand und der Informationsverarbeitung der befragten Person ist. Verschiedene Personen können demselben Ereignis somit unterschiedliche Wahrscheinlichkeiten zuordnen [Eis13]. Betrifft die Einschätzung der Experten jedoch empirische Größen, wie in dieser Arbeit, ist eine valide Annahme, dass sich die angegebenen Wahrscheinlichkeiten zumindest ähneln werden – sofern die Experten einen ausreichend großen, ähnlichen

¹ Hinweis: Auch als Prinzip des unzureichenden Grundes bezeichnet [Lau98].

Erfahrungsschatz besitzen. Denn nach Morgan und Henrion konvergiert eine subjektiv gewählte Wahrscheinlichkeitsverteilung einer empirischen Größe gegen deren frequentistische Wahrscheinlichkeitsverteilung, wenn für die Konstruktion der subjektiven Verteilung mehr und mehr Evidenzen herangezogen werden [Mor90].

Der erfahrene Experte kann in der hier vorgebrachten Argumentation als Beobachter historischer, nicht in einer Datenbasis vorhandenen Evidenzen angesehen werden, die auf dem Wege der Expertenbefragung dennoch implizit in die Wissensbasis eingebracht werden können. Weiterhin existieren in der Literatur diverse Verfahren zur Kombination unterschiedlicher Expertenmeinungen [Jac95].

2.4.2 Wahrscheinlichkeitstheorie

Ein zentrales Element der Wahrscheinlichkeitstheorie ist ein Ereignis (vgl. Abschnitt 2.4.1). Dieses kann beispielsweise gegeben sein durch das Ergebnis eines Glücksrad-Gewinnspiels, wie in Abbildung 2.3 dargestellt.

Es werde der Raum solcher Ergebnisse mit Ω bezeichnet. Für den Ereignisraum S gelten nun die drei Eigenschaften:

1. $\emptyset \in S$ und $\Omega \in S$,
2. Wenn $\alpha, \beta \in S$, dann $\alpha \cup \beta \in S$,
3. Wenn $\alpha \in S$, dann $\Omega \setminus \alpha \in S$.

Nun müssen für eine Abbildung von S nach \mathbb{R} und mithin für eine Wahrscheinlichkeitsverteilung P über (Ω, S) noch die drei Axiome nach Kolmogorow [Kol33] erfüllt sein:

1. $P(\alpha) \geq 0$ für alle $\alpha \in S$,
2. $P(\Omega) = 1$,
3. Wenn $\alpha, \beta \in S$ und $\alpha \cap \beta = \emptyset$, dann $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$.

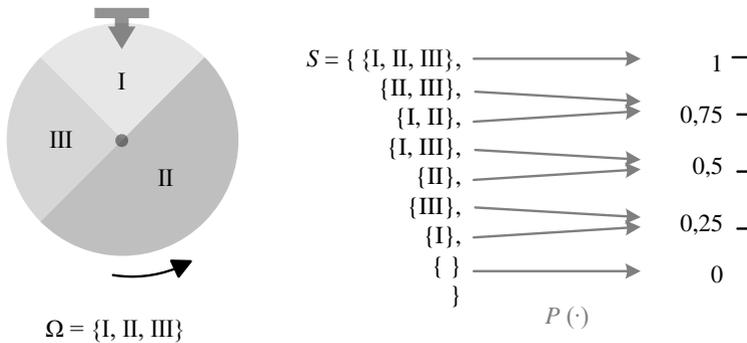


Abbildung 2.3: Glücksrad-Gewinnspiel in Anlehnung an [Bau14].

Als anschauliches Beispiel sei das Glücksrad-Gewinnspiel in Abbildung 2.3 gegeben. Die Flächen des Rades sind mit römischen Ziffern versehen. Der Ergebnisraum ist durch $\Omega = \{I, II, III\}$ gegeben. Die Potenzmenge S möglicher Ereignisse beinhaltet sowohl $\Omega = \{I, II, III\}$ als auch $\emptyset = \{ \}$. Das Ereignis $\alpha = \{II\} \in S$ repräsentiert ein gerades Ergebnis. Es gilt $\Omega \setminus \alpha = \{I, III\} \in S$. Sei $\beta = \{I, III\} \in S$, dann ist ebenso $\alpha \cup \beta = \{I, II, III\} \in S$.

Wenn nun die Wahrscheinlichkeit proportional zur Fläche vergeben und mit der Gesamtfläche normiert wird, dann gilt für Abbildung $P(\cdot)$ zum Beispiel, dass $P(\alpha) = P(\{II\}) = 0,5$. Weil weiterhin $\alpha \cap \beta = \emptyset$, gilt $P(\alpha \cup \beta) = P(\{I, II, III\})$ ist gleich der Summe der einzelnen Abbildungen $P(\alpha) + P(\beta) = P(\{II\}) + P(\{I, III\}) = 1$.

Definition 2.3. Die bedingte Wahrscheinlichkeit ist die Wahrscheinlichkeit des Ereignisses β unter Kenntnis des Eintretens des Ereignisses α :

$$P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)}. \tag{2.1}$$

Wie sich leicht nachrechnen lässt, ergibt sich die bedingte Wahrscheinlichkeit für die konkreten, sich ausschließende Ereignisse, $\alpha = \{II\}$ und $\beta = \{I, III\}$ zu:

$P(\beta|\alpha) = 0$. Das bedeutet, dass sich die a-prior bekannte Wahrscheinlichkeit $P(\beta) = 0,5$ unter Kenntnis des Eintretens des Ereignisses α auf $P(\beta|\alpha) = 0$ reduziert hat.

Sei nun ein weiteres Ereignis $\gamma = \{\text{III}\}$ mit $P(\gamma) = 0,25$ gegeben. Die bedingte Wahrscheinlichkeit $P(\gamma|\beta)$ ergibt sich nach Gleichung (2.1) zu:

$$P(\gamma|\beta) = \frac{P(\beta \cap \gamma)}{P(\beta)} = \frac{P(\{\text{I, III}\} \cap \{\text{III}\})}{P(\{\text{I, III}\})} = \frac{P(\{\text{III}\})}{P(\{\text{I, III}\})} = \frac{0,25}{0,75} = 0,\bar{3}.$$

Die a-priori Wahrscheinlichkeit $P(\gamma) = 0,25$ hat sich also unter Kenntnis des Ereignisses β (d.h. es liegt ein ungerades Ergebnis vor) auf $P(\gamma|\beta) = 0,\bar{3}$ erhöht. Damit ergibt sich die bedingte Wahrscheinlichkeit anschaulich aus dem Verhältnis der Ergebnisse die sowohl in β als auch in γ liegen zu den Ergebnissen die in β liegen [Kol09].

Um die Kettenregel der bedingten Wahrscheinlichkeiten zu erhalten, kann Gleichung 2.1 umgestellt werden zu:

$$P(\alpha \cap \beta) = P(\alpha)P(\beta|\alpha). \quad (2.2)$$

Allgemein gilt:

Definition 2.4. Die Kettenregel der bedingten Wahrscheinlichkeiten ist definiert als:

$$P(\alpha_0 \cap \dots \cap \alpha_k) = P(\alpha_0)P(\alpha_1|\alpha_0) \dots P(\alpha_k|\alpha_0 \cap \dots \cap \alpha_{k-1}).$$

Aus Gleichung 2.2 folgt unter Ausnutzung von

$$P(\alpha \cap \beta) = P(\beta \cap \alpha) \text{ sowie} \\ P(\alpha)P(\beta|\alpha) = P(\beta)P(\alpha|\beta)$$

und Auflösung nach $P(\alpha|\beta)$ die nachfolgende Definition.

Definition 2.5. Der Satz von Bayes ist definiert als

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}.$$

Dieser ergibt sich ebenso über Gleichung 2.1 derart dass,

$$P(\alpha|\beta) = \frac{P(\alpha \cap \beta)}{P(\beta)} = \frac{\frac{P(\alpha \cap \beta)}{P(\alpha)}P(\alpha)}{P(\beta)} = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}.$$

Spruch: Mit dem Satz von Bayes lässt sich aus den a-priori Wahrscheinlichkeiten $P(\alpha)$ und $P(\beta)$ sowie der Likelihood $P(\beta|\alpha)$ die a-posteriori Wahrscheinlichkeit $P(\alpha|\beta)$ berechnen. Das ist besonders dann nützlich, wenn sich in der praktischen Anwendung die a-posteriore Wahrscheinlichkeit $P(\alpha|\beta)$ nicht oder nur unter enormen Aufwand bestimmen lässt.

Weiterhin kann von Interesse sein, wie Ereignisse voneinander abhängen. Für eine vollständige Unabhängigkeit gilt [Kol09]:

Definition 2.6. In P ist ein Ereignis α unabhängig von einem Ereignis β , also $P \models (\alpha \perp \beta)$, wenn $P(\alpha|\beta) = P(\alpha)$ oder wenn $P(\beta) = 0$.

Oder aus einer andere Perspektive betrachtet: P genügt $(\alpha \perp \beta)$ genau dann wenn $P(\alpha \cap \beta) = P(\alpha)P(\beta)$:

$$P(\alpha|\beta) = \frac{P(\alpha \cap \beta)}{P(\beta)} = \frac{P(\alpha)P(\beta)}{P(\beta)} = P(\alpha).$$

Im Beispiel des Glücksrad-Gewinnspiels ist ein Ereignis $\alpha_2 = \{\text{II}\}$ unabhängig von einem Ereignis $\alpha_1 = \{\text{II}\}$ aus einer vorangegangenen Drehung des Rades. Es können aber auch Ereignisse der selben Drehung unabhängig voneinander sein.

Zur Veranschaulichung werde das Glücksrad aus Abbildung 2.3 auf 6 Kreis-segmente zu einem fairen Glücksrad erweitert: $\Omega = \{\text{I}, \text{II}, \text{III}, \text{IV}, \text{V}, \text{VI}\}$ sowie $P(\{\text{I}\}) = P(\{\text{II}\}) = P(\{\text{III}\}) = P(\{\text{IV}\}) = P(\{\text{V}\}) = P(\{\text{VI}\})$. Für dem

fairen Glücksrad-Gewinnspiel gilt nun, dass die zwei Ereignisse der selben Drehung $\alpha = \{I, III, V\}$ und $\beta = \{I, II\}$ unabhängig voneinander sind, denn $P(\alpha|\beta) = P(\alpha)$. Oder anders ausgedrückt: die a-priorie Wahrscheinlichkeit eines ungeraden Ergebnisses $P(\alpha) = 0,5$ bleibt auch bei Kenntnis des Ereignisses β erhalten.

Ereignisse können allerdings auch nur bedingt unabhängig sein. Hierfür muss gelten [Kol09]:

Definition 2.7. In P ist ein Ereignis α bedingt unabhängig von Ereignis β gegeben Ereignis γ , also $P \models (\alpha \perp \beta|\gamma)$, wenn $P(\alpha|\beta \cap \gamma) = P(\alpha|\gamma)$ oder wenn $P(\beta \cap \gamma) = 0$.

Oder alternativ betrachtet: Eine Verteilung P genügt $(\alpha \perp \beta|\gamma)$ genau dann wenn $P(\alpha \cap \beta|\gamma) = P(\alpha|\gamma)P(\beta|\gamma)$.

Gegeben sei wieder das 6-segmentige, faire Glücksrad. Darüber hinaus seien zunächst zwei abhängige Ereignisse gegeben: $\alpha = \{I\}$ und $\beta = \{I, III, V\}$. Gegeben ein weiteres Ereignis $\gamma = \{I, III\}$ sind α und β allerdings bedingt unabhängig: $P(\alpha|\beta \cap \gamma) = P(\{I\} | \{I, III, V\} \cap \{I, III\}) = P(\{I\} | \{I, III\}) = P(\alpha|\gamma)$.

Wie zuvor beschrieben, können Ereignisse anschaulich über Mengen von Ergebnissen beschrieben werden. In der Praxis werden Ergebnisse häufig über Werte referenziert und zu Mengen gebündelt. Bei der fairen Glücksrad-Lotterie könnte dies zum Beispiel die Menge alle ungeraden Ergebnisse sein.

Nach [Kol09] ist die Zuweisung solcher Werte zu einem Ergebnis gegeben durch:

Definition 2.8. Eine Zufallsvariable \underline{X} sei definiert als eine Funktion $F_{\underline{X}}$. Diese Funktion weist jedem Ergebnis $\omega \in \Omega$ einen Wert x zu.

Ein Ereignis kann also über $\{\omega \in \Omega : F_{\underline{X}}(\omega) = x\}$ angegeben werden – oder in Kurzschreibweise $\underline{X} = x$. Dabei wird eine Zufallsvariable \underline{X} als diskret bezeichnet, wenn der Wertebereich $Val(\underline{X})$ von \underline{X} endlich oder abzählbar ist

– sie kann aber auch kontinuierlich sein, beispielsweise reellwertig [Nea04]. Gilt $|\text{Val}(\underline{X})| = 2$, wird die Zufallsvariable \underline{X} in dieser Ausarbeitung als binär bezeichnet.

Im Beispiel der fairen Glücksrad-Lotterie weist eine Zufallsvariable \underline{X} jedem Ergebnis zu, ob dieses ungerade oder gerade ist. Das bedeutet, der Wertebereich der binären Zufallsvariablen ist gegeben durch den Wertebereich $\text{Val}(\underline{X}) = \{\text{ungerade}, \text{gerade}\}$. Ein bestimmtes Ereignis kann zum Beispiel angegeben werden als $\{\omega \in \Omega : F_{\underline{X}}(\omega) = \text{gerade}\} = \{\text{II}, \text{IV}, \text{VI}\}$. In Verbindung mit $P(\cdot)$ ergibt sich die Schreibweise: $P(\underline{X} = \text{gerade}) = P(\{\text{II}, \text{IV}, \text{VI}\})$. Sofern sich aus dem Kontext die Zufallsvariable bereits ergibt, wird diese auch weggelassen, also im Beispiel statt $P(\underline{X} = \text{gerade})$ vereinfachend $P(\text{gerade})$ geschrieben.

2.4.3 Definition

Um im Rahmen einer probabilistischen Wissensbasis eine multivariate Verteilung P über Zufallsvariablen $\underline{X}_{0:n} = \underline{X}_0, \dots, \underline{X}_n$ angeben zu können, muss eine Vielzahl an Parametern festgelegt werden. Selbst in einem einfachen Falle von binären Zufallsvariablen müssten 2^{n+1} mögliche Ergebniskombinationen parametrisiert werden, d.h. $2^{n+1} - 1$ Parameter gesetzt werden. Für 12 binäre Zufallsvariablen ergeben sich also $4096 - 1 = 4095$ Parameter. Offensichtlich steigt die Anzahl der Parameter exponentiell mit der Anzahl der Zufallsvariablen [Phi15b].

Problematisch ist dies zum Beispiel, wenn diese Parameter im Rahmen einer Expertenbefragung erhoben werden sollen. Neben dem rein quantitativen Aspekt, ist die Frage, ob Experten dabei überhaupt sinnvolle Wahrscheinlichkeitswerte für sämtliche Ergebniskombination abschätzen können [Phi16b]. Aber auch wenn Wahrscheinlichkeitswerte aus gegebenen Daten robust gelernt werden sollen, kann dies im Hinblick auf die sehr große Anzahl an Parametern und der dafür notwendigen Datenmenge problematisch sein [Kol09]. Solcherlei Gründe waren zunächst der Hemmschuh für den Einsatz probabilistischer Wissensbasen in früheren Wissensbasierten Systemen bzw. Expertensystemen [Kol09].

Mit Hilfe des Formalismus Bayes-Netz (BN) und der Ausnutzung von Unabhängigkeit zwischen Zufallsvariablen können probabilistische Wissensbasen allerdings kompakter dargestellt und handhabbar repräsentiert werden.

Definition 2.9. Ein BN über den Zufallsvariablen $\underline{X}_{0:n} = \underline{X}_0, \dots, \underline{X}_n$ ist gegeben durch ein Tupel

$$(G, P), \quad (2.3)$$

wobei G einen gerichteten, azyklischen Graphen

$$G = (\mathcal{V}, \mathcal{E}) \quad (2.4)$$

und

$$P(\underline{X}_{0:n}) = \prod_{i=0}^n P(\underline{X}_i | Pa(\underline{X}_i)) \quad (2.5)$$

die gemeinsame Wahrscheinlichkeitsverteilung repräsentieren [Kol09].

Der Graph G wird verwendet, um Abhängigkeiten zwischen Zufallsvariablen $\underline{X}_{0:n}$ zu definieren. Er wird auch als Struktur des BN bezeichnet und äußert sich als gerichteter (engl. directed) azyklischer Graph (DAG). Seine Knoten- bzw. Vertexmenge \mathcal{V} repräsentiert die Zufallsvariablen $\underline{X}_{0:n}$.

Während eine gerichtete Kante $v_i \rightarrow v_j$ der Kantenmenge \mathcal{E} eine direkte Abhängigkeit zwischen zwei Variablen angibt, symbolisiert eine fehlende Kante die Unabhängigkeit zweier Variablen¹.

¹ Hinweis: Da Graph G sowohl gerichtet als auch azyklisch ist, existiert kein Pfad der von einem Knoten zu sich selbst zurück führt: \nexists Pfad $v_i \rightarrow \dots \rightarrow v_i$.

Die gemeinsame Wahrscheinlichkeitsverteilung ist durch das Produkt aller bedingten Wahrscheinlichkeitsverteilungen gegeben, welche mit den Knoten von G verknüpft sind. Sie repräsentiert die Parameter des BN. Dabei bezeichnet $Pa(\underline{X}_i)$ die Menge der Eltern¹ einer Zufallsvariablen \underline{X}_i . Im Graphen entspricht dies Knoten mit einer gerichteten Kante zum Knoten der mit \underline{X}_i verknüpft ist². Die Faktoren $P(\underline{X}_i | Pa(\underline{X}_i))$ werden als Conditional Probability Distribution – bedingte Wahrscheinlichkeitsverteilung (CPD) bezeichnet. Im Falle diskreter Zufallsvariablen können sogenannte Conditional Probability Tables – bedingte Wahrscheinlichkeitstabellen (CPT) herangezogen werden, um die Verteilungen zu repräsentieren.

Bei der obigen Darstellung wird angenommen, dass G eine I-MAP (engl. Independence Map) der Wahrscheinlichkeitsverteilung P ist, d.h. dass jede Unabhängigkeitsbeziehung, die durch G gegeben ist, auch in P erfüllt ist³ [Kol09]. Nur in diesem Falle kann die Berechnung von P durch die bedingten Unabhängigkeiten so vereinfacht werden, dass sich die „Kettenregel der Bayesschen Netze“, wie in Formel (2.5) dargestellt, ergibt.

Um die für die I-MAP benötigten bedingten Unabhängigkeiten formal angeben zu können, müssen folgende Knotenmengen definiert werden:

- $Pa(\underline{X}_i)$: Menge der Knoten, von denen eine gerichtete Kante nach \underline{X}_i führt (d.h. die Eltern von \underline{X}_i),
- $Desc(\underline{X}_i)$: Menge der Knoten, die auf einem von \underline{X}_i ausgehenden Pfad liegen (d.h. die Nachfahren von \underline{X}_i),
- $NDesc(\underline{X}_i)$: Menge der Knoten, die weder in $Pa(\underline{X}_i)$ noch in $Desc(\underline{X}_i)$ liegen, sowie nicht \underline{X}_i selbst sind (d.h. Nicht-Nachfahren von \underline{X}_i).

¹ Hinweis: Durch den verwendeten DAG (mit seiner azyklischen sowie gerichteten Eigenschaft) ist sichergestellt, dass die Elternbeziehung eindeutig für jeden Knoten angegeben werden kann.

² Hinweis: Wenn $Pa(\underline{X}_i) = \emptyset$, ist die Zufallsvariable \underline{X}_i ein Wurzelknoten und $P(\underline{X}_i | \emptyset) = P(\underline{X}_i)$ gibt folglich die A-priori-Wahrscheinlichkeit an.

³ Hinweis: Umgekehrt müssen aber nicht alle Unabhängigkeiten in P auch in G repräsentiert sein.

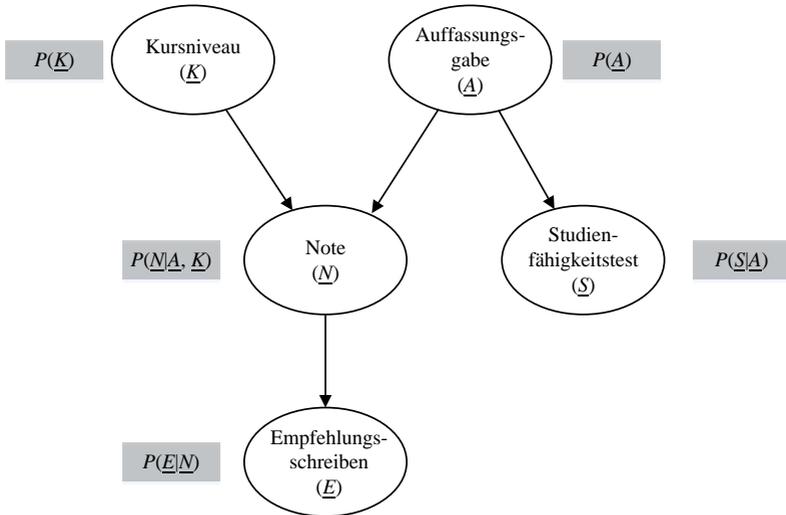


Abbildung 2.4: Bayessches Netz aus [Kol09, Kor10, Dar09] in Anlehnung an [Pea88]. Graph mit Knoten und zugeordneten bedingten Wahrscheinlichkeitsverteilungen. Es gilt: $P(\underline{A}, \underline{E}, \underline{K}, \underline{N}, \underline{S}) = P(\underline{A})P(\underline{E}|\underline{N})P(\underline{N}|\underline{A}, \underline{K})P(\underline{K})P(\underline{S}|\underline{A})$.

Mit diesen Knotenmengen kann die lokale Markovbedingung, die der I-MAP zugrunde liegt, definiert werden:

Definition 2.10. Der DAG G eines Bayes-Netztes, dessen Knoten v die Zufallsvariablen $\underline{X}_{0:n}$ repräsentieren, steht für die Menge der lokalen bedingten Unabhängigkeiten

$$I_l(G) = \{\underline{X}_i \perp \text{NDesc}(\underline{X}_i) | \text{Pa}(\underline{X}_i)\} : i = 0 \dots n \quad (2.6)$$

Die lokale Markovbedingung besagt also, dass jeder Knoten (\underline{X}_i) bedingt unabhängig von nicht-nachfolgenden Knoten ist, gegeben seine Eltern, vgl. [Kol09].

Beispielhaft sei das Bayes-Netz in Abbildung 2.4 gegeben. Sein DAG stellt Knoten und die jeweils verknüpften Zufallsvariablen aus der Menge

$\{\underline{A}, \underline{E}, \underline{K}, \underline{N}, \underline{S}\}$ dar. Aus didaktischen Gründen tragen die Knoten zusätzlich eine sprechende Bezeichnung. Deutet man die Kanten in einem kausalen Zusammenhang (Ursache und Wirkung), so können sowohl das Kursniveau als auch die Auffassungsgabe eines Studenten seine Note beeinflussen. Die Note wiederum beeinflusst die Qualität des Empfehlungsschreibens. Die Auffassungsgabe beeinflusst zudem das Ergebnis des Studienfähigkeitstest (engl. Scholastic Assessment Test, SAT)¹. Auch fehlende Kanten sind bedeutungstragend: Die nicht vorhandene Kante zwischen dem Knoten Kursniveau und dem Knoten Auffassungsgabe zeigt an, dass die verknüpften Zufallsvariablen \underline{K} und \underline{A} unabhängig voneinander sind.

Durch den DAG in Abbildung 2.4 ist die Menge der lokalen bedingten Unabhängigkeiten wie folgt festgelegt:

$$I_l(G) = \{(\underline{E} \perp \underline{A}, \underline{K}, \underline{S} | \underline{N}), (\underline{S} \perp \underline{E}, \underline{K}, \underline{M} | \underline{A}), (\underline{N} \perp \underline{S} | \underline{A}, \underline{K}), \\ (\underline{K} \perp \underline{A}, \underline{S} | \emptyset), (\underline{A} \perp \underline{K} | \emptyset)\}.$$

2.4.4 d-Separation

In Definition 2.10 aus Kapitel 2.4.3 wurde gezeigt, dass durch den Graphen $G = (V, E)$ eines Bayes-Netzes die Menge lokaler bedingter Unabhängigkeiten $I_l(G)$ gegeben ist.

Darüber hinaus kann ein Graph weitere, sogenannte globale Markovunabhängigkeiten enthalten. Diese lassen sich mittels des Konzeptes der d-Separation prüfen [Gei90]. Dieses Konzept kann genutzt werden, um zu überprüfen ob zwei Mengen an Zufallsvariablen gegeben einer Menge anderer Zufallsvariablen unabhängig von einander sind.

Um dieses Konzept herzuleiten wird zunächst der Begriff des (aktiven) Weges definiert, vgl. [Gei90] und [Kol09]:

¹ Hinweis: Eine detaillierte Darstellung dieses Beispiels findet sich auch in der einschlägigen Literatur [Kol09, Kor10, Pea88].

Definition 2.11. Gegeben sei der Graph $G = (\mathcal{V}, \mathcal{E})$ eines Bayes-Netzes mit Zufallsvariablen $\underline{X}_{0:n}$. Weiterhin sei \mathcal{Z} eine Teilmenge der Zufallsvariablen, die beobachtet wurde. \mathcal{Z} sei durch die Knotenmenge $\mathcal{V}_{\mathcal{Z}} \subseteq \mathcal{V}$ repräsentiert. Ein Weg ist eine Sequenz von Kanten, die in der ungerichteten Repräsentation des DAGs $G = (\mathcal{V}, \mathcal{E})$ einen Pfad (Sequenz von Knoten) bildet¹. Hierfür schreibt man auch $v_0 \rightleftharpoons \dots \rightleftharpoons v_m$ mit $v_0, \dots, v_m \in \mathcal{V}$.

Ein solcher Weg wird als aktiver Weg gegeben \mathcal{Z} bezeichnet, falls:

1. Für jede existierende konvergierende Verbindung $v_{i-1} \rightarrow v_i \leftarrow v_{i+1}$ innerhalb des Weges, liegen entweder v_i oder einer seiner Nachfolger in $\mathcal{V}_{\mathcal{Z}}$.
2. Kein anderer Knoten des Weges liegt in $\mathcal{V}_{\mathcal{Z}}$.

Insbesondere ist der Weg $v_0 \rightleftharpoons \dots \rightleftharpoons v_m$ nicht aktiv, wenn v_0 oder v_m in $\mathcal{V}_{\mathcal{Z}}$ liegen.

Mithilfe der Definition eines aktiven Weges kann das Konzept der d-Separation definiert werden[Gei90]:

Definition 2.12. Seien \mathcal{W} , \mathcal{Y} und \mathcal{Z} drei disjunkte Teilmengen der Zufallsvariablen, die in der Struktur $G = (\mathcal{V}, \mathcal{E})$ eines Bayes-Netzes durch die disjunkten Knotenmengen $\mathcal{V}^{\mathcal{W}}$, $\mathcal{V}^{\mathcal{Y}}$ und $\mathcal{V}^{\mathcal{Z}}$ repräsentiert werden. Dann heißt \mathcal{W} d-separiert von \mathcal{Y} gegeben \mathcal{Z} genau dann, wenn es keinen aktiven Weg zwischen beliebigen Knoten $v^{\mathcal{W}} \in \mathcal{V}^{\mathcal{W}}$ und $v^{\mathcal{Y}} \in \mathcal{V}^{\mathcal{Y}}$ gibt, gegeben $\mathcal{V}^{\mathcal{Z}}$.

Mit dem Konzept der d-Separation können die globalen Markovunabhängigkeiten beschrieben werden.

¹ Hinweis: Die Begriffe Weg und Pfad werden in der deutschsprachigen Literatur teils synonym verwendet. Hier sind die Übersetzungen der Begriffe trail (= Sequenz von Kanten) und path (= Sequenz von Knoten) gemeint, die sich unterscheiden.

Definition 2.13. Sei G der DAG eines Bayes-Netzes und \mathcal{W} , \mathcal{Y} und \mathcal{Z} disjunkte Teilmengen der durch die Knoten des DAG repräsentierten Zufallsvariablen. Dann sind die globalen Markovunabhängigkeiten von G definiert als:

$$I(G) = \{(\mathcal{W} \perp \mathcal{Y} | \mathcal{Z}) : \mathcal{W} \text{ ist d-separiert von } \mathcal{Y} \text{ gegeben } \mathcal{Z}.\} \quad (2.7)$$

Für das Beispiel in Abbildung 2.4 sind somit die globalen Markovunabhängigkeiten gegeben durch:

$$I(G) = \{(\underline{E} \perp \underline{A}, \underline{K}, \underline{S} | \underline{N})\}.$$

2.4.5 Dynamische Aspekte

Ein BN ist ein probabilistisches graphisches Modell (PGM), das graphentheoretische Ansätze mit Ansätzen der Wahrscheinlichkeitstheorie verbindet. Wie in Kapitel 2.4.3 ausführlich erläutert, ist ein BN über den Zufallsvariablen $\underline{X}_{0:n} = \underline{X}_0, \dots, \underline{X}_n$ gegeben durch ein Paar

$$B = (G, P). \quad (2.8)$$

Dabei ist G ein gerichteter, azyklischer Graph

$$G = (\mathcal{V}, \mathcal{E}) \quad (2.9)$$

und

$$P(\underline{X}_{0:n}) = \prod_{i=0}^n P(\underline{X}_i | Pa(\underline{X}_i)) \quad (2.10)$$

die gemeinsame Wahrscheinlichkeitsverteilung [Kol09].

Der Graph G wird genutzt, um Abhängigkeiten zwischen den Zufallsvariablen zu definieren und repräsentiert damit die Struktur des BN (vgl. Kapitel 2.4.3). Die Vertexmenge \mathcal{V} repräsentiert die Menge der Zufallsvariablen, während

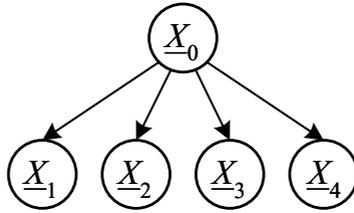


Abbildung 2.5: Graph eines Bayes-Netzes (BN) über den Zufallsvariablen $\underline{X}_{0:4}$. Diese Struktur, die auch Naive BN genannt wird, ist gekennzeichnet durch gerichtete Kanten ausgehend von Wurzelknoten \underline{X}_0 zu seinen Kindern $\underline{X}_{1:4}$. Da eine fehlende Kante die Unabhängigkeit zweier Zufallsvariablen verkörpert, kann die Verbundwahrscheinlichkeit $P(\underline{X}_{0:4})$ wie folgt faktorisiert werden: $P(\underline{X}_{0:4}) = P(\underline{X}_0)P(\underline{X}_1|\underline{X}_0)P(\underline{X}_2|\underline{X}_0)P(\underline{X}_3|\underline{X}_0)P(\underline{X}_4|\underline{X}_0)$.

eine gerichtete Kante $v_i \rightarrow v_j$ der Menge der Kanten \mathcal{E} eine direkte Abhängigkeit zwischen zwei Variablen darstellt. Eine fehlende Kante symbolisiert die Unabhängigkeit dieser beiden Variablen (cf. Figure 2.5).

Die gemeinsame Wahrscheinlichkeitsverteilung repräsentiert die Parameter des BN (vgl. Kapitel 2.4.3) und ist durch das Produkt aller bedingten Wahrscheinlichkeitsverteilungen gegeben, die mit den Vertices von G verknüpft sind. Dabei bezeichnet $Pa(\underline{X}_i)$ die Menge der Eltern einer Zufallsvariablen \underline{X}_i .

Ein Dynamisches Bayes-Netz (DBN) ist eine Erweiterung eines BN, das auch die zeitlichen Abhängigkeiten der Zufallsvariablen berücksichtigen kann [Mur02]. Ein DBN ist dabei gegeben durch das Tupel

$$DBN = (B_0, B_{\rightarrow}).$$

Ersteres, also B_0 , wird genutzt, um die Struktur als auch die a-priorie Wahrscheinlichkeitsverteilung über den Zufallsvariablen $\underline{X}_{0:n}$ in Zeitschritt 0, sprich: $P(\underline{X}_{0:n}^0)$, anzugeben. Die Struktur von B_0 kann beispielsweise nach Abbildung 2.5 erfolgen – die Variablen tragen dann zusätzlich einen hochgestellten Index des Zeitschritts 0.

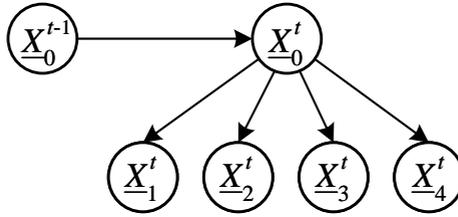


Abbildung 2.6: Vereinfachte Darstellung eines zeitlichen Bayes-Netztes mit zwei Zeitscheiben. Dieses wird im englischsprachigen Raum auch als Temporal Bayesian Network (TBN) mit zwei Zeitscheiben oder kurz als 2TBN bezeichnet. Das 2TBN (oder auch: B_{\rightarrow}) wird als Schablone für nachfolgende Zeitschritte t eingesetzt. Aus Gründen der Übersichtlichkeit sind Knoten der Zufallsvariablen $X_{1:4}^{t-1}$ nicht abgebildet, denn es existiert in diesem Beispiel keine direkte Abhängigkeit zu $X_t^{0:4}$. Eine solch vereinfachte Darstellung eines 2TBN, also eine Darstellung mit nur relevanten Knoten des vorgelagerten Zeitschritts, wird teils als 1,5TBN bezeichnet [Mur02].

Weiterhin spezifiziert B_{\rightarrow} die bedingte Wahrscheinlichkeitsverteilung über diskrete Zeitschritte t hinweg, nämlich durch die Verwendung von:

$$P(X_{0:n}^t | X_{0:n}^{t-1}) = \prod_{i=0}^n P(X_i^t | Pa(X_i^t)). \quad (2.11)$$

Dabei bezeichnet $Pa(X_i^t)$ die Menge der Eltern von X_i^t im zugehörigen Graphen. Eltern können hierin dem selben Zeitschritt t oder aber dem vorangegangenen Zeitschritt $t - 1$ zugeordnet sein. Im letzteren Falle zeigen gerichteten Kanten in Zeitschritte mit höherem Index, was das zeitliche Voranschreiten widerspiegelt [Mur02]. Abbildung 2.6 zeigt hierzu beispielhaft den Graphen B_{\rightarrow} eines DBNs. Dabei besitzt der Knoten der Zufallsvariablen X_0^t einen Elternknoten im vorgelagerten Zeitschritt $t - 1$; es existiert also eine gerichtete Kante über zwei verschiedene Zeitschritte hinweg. Weiterhin existieren Kanten innerhalb eines Zeitschrittes, beispielsweise zwischen Knoten der Zufallsvariablen X_0^t und X_1^t .

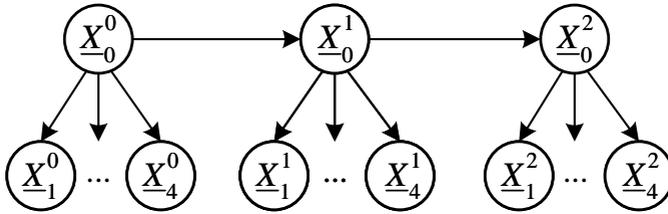


Abbildung 2.7: Die Abbildung zeigt ein über 3 Zeitschritte ausgerolltes DBN. Zur Vereinfachung sind einige Knoten nicht dargestellt, was durch \dots veranschaulicht wird.

Die gemeinsame Wahrscheinlichkeitsverteilung über mehrere Zeitschritte hinweg ergibt sich als [Mur02]:

$$P(\underline{X}_{0:n}^{0:m}) = \left(\prod_{i=0}^n P(\underline{X}_i^0 | Pa(\underline{X}_i^0)) \right) \prod_{t=1}^m \prod_{i=0}^n P(\underline{X}_i^t | Pa(\underline{X}_i^t)) \quad (2.12)$$

$$= \prod_{t=0}^m \prod_{i=0}^n P(\underline{X}_i^t | Pa(\underline{X}_i^t)). \quad (2.13)$$

Anschaulich entspricht das einem Ausrollen des DBNs. Hierbei fungiert B_0 als Startverteilung für Zeitschritt 0 und B_{\rightarrow} als Schablone für alle weiteren Zeitschritte, vgl. hierzu auch Gleichung (2.12) sowie Abbildung 2.7. Letztere zeigt den Graph eines DBN, der mit $m = 2$ ausgerollt wurde.

2.5 FHIR zur Wissensrepräsentation

Ziel der FHIR¹ ist die Unterstützung des Austauschs von Informationen zwischen Softwaresystemen² im medizinischen Bereich³. Herausgeber ist die Health Level Seven (HL7)⁴ International, welche im Jahr 1987 als non-Profit Organisation gegründet wurde und durch das American National Standards Institute (ANSI) seit 27 Jahren als Normungsorgan akkreditiert ist. Gemeinsam mit der ISO und anderen Normungsorganen werden kontinuierlich international einschlägige Normen etabliert.

Mit FHIR können sowohl Ressourcen als auch Datenformate und Schnittstellen spezifiziert werden [Ben21]. Ressourcen sind allerdings die zentralen Bausteine der FHIR [Oem21]. Aus diesen allgemeinen Schablonen können domänentypische Objekte wie beispielsweise Patient (engl. Patient), Untersuchungswert (engl. Observation) oder Behandlungsort (engl. Location) abgeleitet werden. Es stehen 145 verschiedene Ressourcen zur Verfügung [FHI21], die sich gemäß ihres Typus in einer Taxonomie verorten lassen [Ben21]. Mit Berücksichtigung der entsprechenden fachlichen Sicht ergibt sich Tabelle 2.2 auf Seite 36 mit den jeweils beispielhaft dargestellten Ressourcen.

Weitere Ressourcen können nach Bedarf hinzugefügt werden [Ben21]. Insgesamt können so Daten mittels FHIR auffindbar (engl. Findable), zugänglich

¹ Gesprochen: [ˈfaɪə] (engl. fire).

² Hinweis: Systeme werden in der Begriffswelt der Softwaretechnik als Ausschnitte der realen oder gedanklichen Welt gesehen, welche aus Komponenten, darauf vorhandenen Strukturen sowie einer Systemgrenze bestehen [Hes84, Bal10]. Letztere wird auch als Schnittstelle oder seltener als Nahtstelle bezeichnet (engl. interface).

³ Hinweis: Der Anwendungsbereich umfasst Human- und Veterinärmedizin und hierin die medizinische Versorgung selbst als auch Finanz- und Verwaltungsaspekte [Ben21, FHI21].

⁴ Hinweis: Die Zahl 7 ist eine Anspielung auf das Open Systems Interconnection (OSI) Schichtenmodell der International Organization for Standardization (ISO) – häufig kurz als ISO/OSI Modell bezeichnet [Pro93]. Dieses bezieht sich auf die Verbindung (Interconnection) verschiedener Systeme. Hierin ist die Schicht 7 die sog. Anwendungsschicht, welche sich mit dem Austausch von Daten auf Anwendungsebene beschäftigt.

(engl. Accessible), interoperabel (engl. Interoperable) sowie wiederverwendbar (engl. Reusable) genutzt werden – also sprich: nach dem sogenannten FAIR-Prinzip [Wil16, Thu21].

2.5.1 Motivation

FHIR gewann in den letzten Jahren zunehmend an Bedeutung [Leh19a, Hub20]. Schon im Jahr 2018 verständigten sich die Technologieunternehmen IBM, Microsoft, Oracle, Google, Amazon und Salesforce in einer gemeinsamen Erklärung auf die Notwendigkeit bestehende Hemmnisse bei der medizinischen Datenrepräsentation zu beseitigen. Hierin werden FHIR und angeschlossene Projekte bereits explizit genannt [Inf18, Leh19a], denn sie sollen es ermöglichen, medizinische Daten mit Methoden der KI weltweit sowie unternehmensübergreifend effektiv und effizient nutzen zu können [Leh19b].

Mittlerweile wurden auf Basis dieser Bestrebungen Googles Cloud Healthcare API [Goo21] und Microsofts Azure API FHIR [Mic21] als Dienste zur Analyse großer medizinischer Datenbestände realisiert [Hub20]. Bei letzterem Dienst werden in regelmäßigen Abständen Fallbeispiele der Nutzer dieses Dienstes aus dem universitären und unternehmerischen Bereich veröffentlicht [Mic21].

Im Bereich der Software für Patienten nutzt das Unternehmen Apple im Zuge der Verknüpfung von elektronischen Gesundheitsdaten mit der eigenen Health App die FHIR Technologie beispielsweise für die Repräsentation von Allergien, Impfungen, Medikamenten und Vitalwerten [App21]. Solcherlei Umsetzungen mit FHIR werden auch allgemeiner betrieben. Zum Beispiel mit den Substitutable Medical Applications Reusable Technologies (SMART), die einen Einstiegspunkt für die Entwicklung von Gesundheitsanwendungen bieten [Man16] und die aus der Zusammenarbeit zwischen dem Bostoner Children's Hospital und der Harvard Medical School entstanden sind.

Weiterhin ergab sich speziell in Deutschland die Frage nach akzeptierbaren Softwarelösungen im Hinblick auf ePA, deren Entwicklung durch entsprechende Gesetze in den Jahren zuvor angeregt wurden. Im Jahr 2019 stellten

Tabelle 2.2: In FHIR stehen mit Version 4 R4 insgesamt 145 verschiedene Ressourcen zur Verfügung [FHI21]. Diese lassen sich gemäß ihres Typus in einer Taxonomie verorten [Ben21], die sich auf fachliche Sichten herunterbrechen lässt. Die erste Spalte der Tabelle zeigt die fachlichen Sichten und jeweils eine beispielhaft dargestellte Ressource in der zweiten Spalte. Es existiert eine große Bandbreite an Sichten und somit auch an zugeordneten Ressourcen.

Sicht	Beispielressource
Arzneimittel	<i>Medication</i>
Bildgebung	<i>ImagingStudy</i>
Biomedizinische Forschung	<i>ResearchStudy</i>
FHIR Infrastruktur	<i>Resource</i>
Finanzen	<i>Invoice</i>
Gemeinschaftliche Pflege	<i>Consent</i>
Klinische Genomik	<i>MolecularSequence</i>
Klinische Qualitätsinformationen	<i>MeasureReport</i>
Medizinische Entscheidungsunterstützung	<i>PlanDefinition</i>
Medizinische Geräte	<i>DeviceMetric</i>
Nachrichtenaustausch	<i>MessageDefinition</i>
Öffentliche Gesundheit	<i>Immunization</i>
Patientenversorgung	<i>Goal</i>
Patientenverwaltung	<i>Patient</i>
Sicherheit	<i>Provenance</i>
Strukturdokumente	<i>DocumentManifest</i>
Verordnungen	<i>Substance</i>
Vokabular	<i>ValueSet</i>

hierzu [Plo19] ein Konzept vor, welches FHIR nutzt, um durch eine möglichst akzeptierbare Lösung eine hohe Akzeptanz bei den Benutzern zu erreichen. Heute, im Jahr 2022, basieren Schnittstellen sowie die auf der ePA ablegbaren und in Bearbeitung befindlichen MIO wie beispielsweise Impfpass, Mutterpass oder das Zahnärztliche Bonusheft auf FHIR [Kas21].

2.5.2 Hierarchie der Ressourcen und Elemente

Neben Ressourcen existieren in FHIR sogenannte Elemente, welche zum Befüllen der Ressourcen dienen. Elemente sind beispielsweise primitive Datentypen (Ganzzahl *integer*, Zeichenkette *string* etc.) oder auch spezielle Datentypen (Postadresse *Address*, Alter *Age* etc.). Bei ersteren kann direkt der Datentyp mitsamt Wert und Bezeichner festgelegt werden (z. B. Datentyp *integer* mit Wert 3 und Bezeichner *anzahlKinder*). Bei letzterem sind je nach Datentyp weitere Klassen für die Spezifikation von Werten und Bezeichnern notwendig.

Gemeinsame Basis von Elementen als auch von Ressourcen in FHIR ist aber ein abstraktes (engl. *abstract*) Wurzelement, von dem ausgehend sich zwei inhaltlich verschiedene Hierarchien aufspannen [Oem21]. Abbildung 2.8 zeigt ein stark vereinfachtes Klassendiagramm zur Verdeutlichung der groben Zusammenhänge am prototypischen Beispiel der Ressource *Patient* und den wichtigsten Datentypen.

Weißer Pfeilspitzen in Abbildung 2.8 zeigen (wie bei Klassendiagrammen üblich) Vererbungen an, wobei der jeweilige Pfeil von der Spezialisierung hin zur Generalisierung zeigt. Die schwarze Raute repräsentiert eine Komposition, also eine Teil-Ganze Beziehung bei der das Teil nicht ohne das Ganze existieren kann. Kardinalitäten, also Kann- bzw. Mussbeziehungen, sind in eckigen Klammern angegeben und beschränken sich in dem dargestellten Ausschnitt auf einfache Kannbeziehungen [0..1] oder mehrfache Kannbeziehungen [0..*].

Auf der rechten Seite in Abbildung 2.8 befinden sich u.a. die von einem abstrakten Element abgeleiteten primitiven Datentypen (beispielsweise Ganzzahl) und allgemeine Datentypen (beispielsweise Adresse). In letzterem lassen

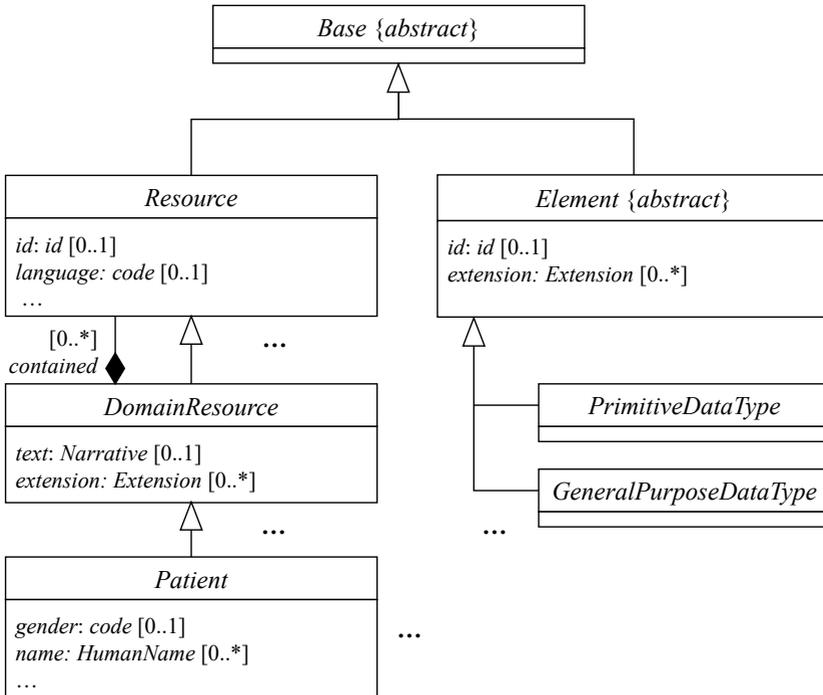


Abbildung 2.8: Die Abbildung zeigt ein Klassendiagramm zur Verdeutlichung der Zusammenhänge am Beispiel der FHIR Ressource *Patient* (links) und den wichtigsten Datentypen (rechts) [FHI21, Oem21]. Zur Vereinfachung sind einige Klassen und Zusammenhänge nicht dargestellt, was durch das Symbol ... veranschaulicht wird. Die getrennten Hierarchien erlauben eine flexible Modellierung der Ressourcen- bzw. Elementaspekte, was als Anlehnung an die aus der Softwareentwicklung als Entwurfsmuster „Brücke“ bekannten Herangehensweise gesehen werden kann. Das gemeinsame Wurzelement *Base* dient zur Etablierung gemeinsamer Eigenschaften die dann sowohl bei Elementen als auch Ressourcen immer vorhanden sein müssen.

sich auch umfangreiche Vokabularien verknüpfen, sodass sich Logical Observation Identifiers Names and Codes (LOINC) oder Systematized Nomenclature of Medicine (SNOMED) in einfacher und konsistenter Weise einbinden lassen.

Auf der linken Seite in Abbildung 2.8 sind die Ressourcen verortet. Konkret wird zunächst die Klasse *Resource* abgeleitet, von der wiederum die Klasse der sogenannten *DomainResource* spezialisiert wird. Diese zusätzliche Abstraktionsebene ermöglicht es, in einer *DomainResource* mehrere Ressourcen zu verankern (engl. contained). Weiterhin lassen sich durch diese zusätzliche Abstraktionsebene flexibel Erweiterungen (engl. Extensions) einbinden. Spezialisierungen der *DomainResource* sind dann beispielsweise *Patient* oder eine der anderen 145 Ressourcen (vgl. Symbole \cdots rechts neben *Patient* in Abbildung 2.8). Die Ressourcen selbst stehen typischerweise auch in Beziehung zueinander, so kann ein *Patient* mehrere behandelnde Ärzte haben, die selbst auch Ressourcen im FHIR'schen Sinne sind.

2.5.3 Medizinische Entscheidungsunterstützung

Für die nachfolgenden Betrachtungen ist aus den unterschiedlichen Sichten (vgl. Tabelle 2.2 auf Seite 36) die Sicht der medizinischen Entscheidungsunterstützung von besonderer Bedeutung. Hierin sind Ressourcen wie *PlanDefinition* und *Action* verortet, die für eine Repräsentation eines Prozessmodells notwendig sind.

Abbildung 2.10 auf Seite 42 zeigt ein Klassendiagramm mit den Klassen *PlanDefinition*, *Action* und *RelatedAction*. Zur besseren Übersichtlichkeit wurden optionale Attribute und Verbindungen zu anderen Klassen vernachlässigt. Zentrales Element ist die *Action*, welche den Baustein einer *PlanDefinition* bildet. Diese Teil-Ganze Beziehung zwischen *PlanDefinition* und *Action* wird durch die Komposition (schwarze Raute) gekennzeichnet. Eine *PlanDefinition* kann ihrerseits aus beliebig vielen Instanzen einer *Action* bestehen, was durch die mehrfache Kannbeziehung [0..*] angegeben wird.

Eine *Action* kann wiederum mit beliebig vielen (*related-*)*Actions* in Beziehung stehen. Hierfür muss die *Action* jeweils die *id* der Zielaktion der Beziehung kennen. Eine solche Beziehung kann neben der expliziten Darstellung über eine solche *id* auch durch eine gerichtete Kante angegeben werden. Die gerichtete Kante zeigt dann von der betrachteten *Action* zur in Beziehung stehenden *Action*. Das bedeutet, es kann von der Quell- zur Zielaktion navigiert werden.

Abbildung 2.9 zeigt die Zusammenhänge an einem Beispiel, in dem jeweils eine zeitliche Beziehungen (*relationship*=before-end) zwischen Aktion *a* und Aktion *b* zu einer anonymen Aktion besteht¹. Zur Übersichtlichkeit wurde an den beiden Kanten in der Abbildung 2.9 auf die Attributnamen *relationship* verzichtet und nur deren Ausprägungen dargestellt.

Im rechten Teil des Klassendiagramms in Abbildung 2.10 auf Seite 42 zeigt sich eine reflexive Komposition. Diese gibt an, dass eine *Action* aus beliebig vielen weiteren *Actions* bestehen kann (aber nicht muss). Diese Beziehungen in Kombination mit zwei Attributen, dem sogenannten *groupingBehavior* und dem sogenannten *selectionBehavior*, ermöglichen, dass beispielsweise nur eines der Kinder ausgeführt werden kann (*groupingBehavior* = logical-group, *selectionBehavior* = exactly-one), höchstens eines der Kinder ausgeführt werden kann (*groupingBehavior* = logical-group, *selectionBehavior* = at-most-one) oder alle Kinder in beliebiger Reihenfolge ausgeführt werden müssen (*groupingBehavior* = logical-group, *selectionBehavior* = all). Diese Festlegungen sind für Verschaltungen von *Actions* wichtig².

Abbildung 2.9 zeigt beispielhaft wie die Aktionen *a* und *b* selektiv ausgeführt werden. In gleicher Weise führt das Setzen einer *selectionBehavior* mit dem Wert all oder aber at-most-one zu entsprechend anderem Verhalten. Die Kinderknoten können dabei selbst Beziehungen zu weiteren Aktionen aufweisen oder auch Elternknoten sein.

Ein weiteres Attribut einer *Action* ist das sogenannte *cardinalityBehavior*. Hierdurch wird festgelegt, ob eine Aktion nur einmalig ausgeführt werden kann (*cardinalityBehavior* = single) oder eine mehrfache Ausführung der

¹ Hinweis: Die Abbildung zeigt ein Objektdiagramm. D.h. im Gegensatz zu einem Klassendiagramm werden nicht Klassen, sondern konkrete Objekte (also Instanzen von Klassen) und ihre Beziehungen zueinander angegeben. Im vorliegenden Fall werden somit die Objekte *a* und *b* angegeben, welche Instanzen der Klasse *Action* sind. Dieser Sachverhalt wird mit *a:Action* bzw. *b:Action* ausgedrückt. Vereinfachend formuliert handelt es sich bei *a* und *b* um Aktionen – oder genauer: um Instanzen der Klasse *Action*. Instanzen ohne Namen (anonyme Objekte) werden mit *:Action* angegeben.

² Die verschiedenen Arten möglicher Verschaltungen werden im folgenden Kapitel 3 (Prozessmodelle) ab Seite 47 näher beleuchtet.

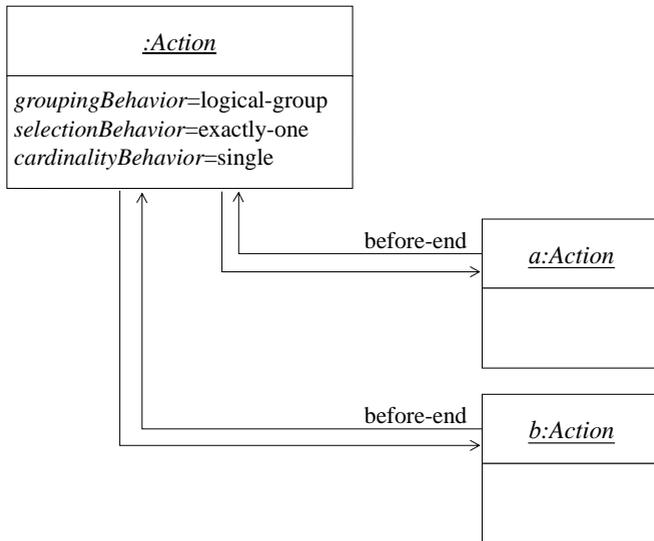


Abbildung 2.9: Die Abbildung zeigt die zwei Aktionen *a*, *b*. Hierbei werden *a* und *b* selektiv ausgeführt – d. h. entweder Aktion *a* oder aber Aktion *b* werden ausgeführt. Ein Knoten des Typs *Action* bündelt die Entscheidung (*groupingBehavior*=logical-group) über das Attribut *selectionBehavior*=exactly-one. Die verbundenen Kindknoten *a* und *b*, von denen nur exakt einer ausgeführt werden kann, müssen beendet sein (*groupingBehavior*=before-end) bevor die Elternaktion endet und mithin die Ausführung andere Aktion beginnen kann. Eine mehrfache Ausführung des Elternknotens ist ausgeschlossen (*cardinalityBehavior*=single)

Aktion möglich ist (*cardinalityBehavior* = multiple). In Abbildung 2.9 wird eine einmalige Ausführung der Entscheidung festgelegt.

Gedanklich zurückkehrend auf das Klassendiagramm in Abbildung 2.10 (Seite 42), kann eine *Action* auch einen (Daten-)input sowie einen (Daten-)output aufweisen.

Letztgenannte sind von der Klasse *DataRequirement*, deren Typ spezifiziert werden muss und vom speziellen Datentypen *Observation* stammen kann. Der Datentyp *Observation* wird in FHIR verwendet, um Messwerte und einfache Angaben unter anderem über Patienten, Geräte, etc. festzuhalten.

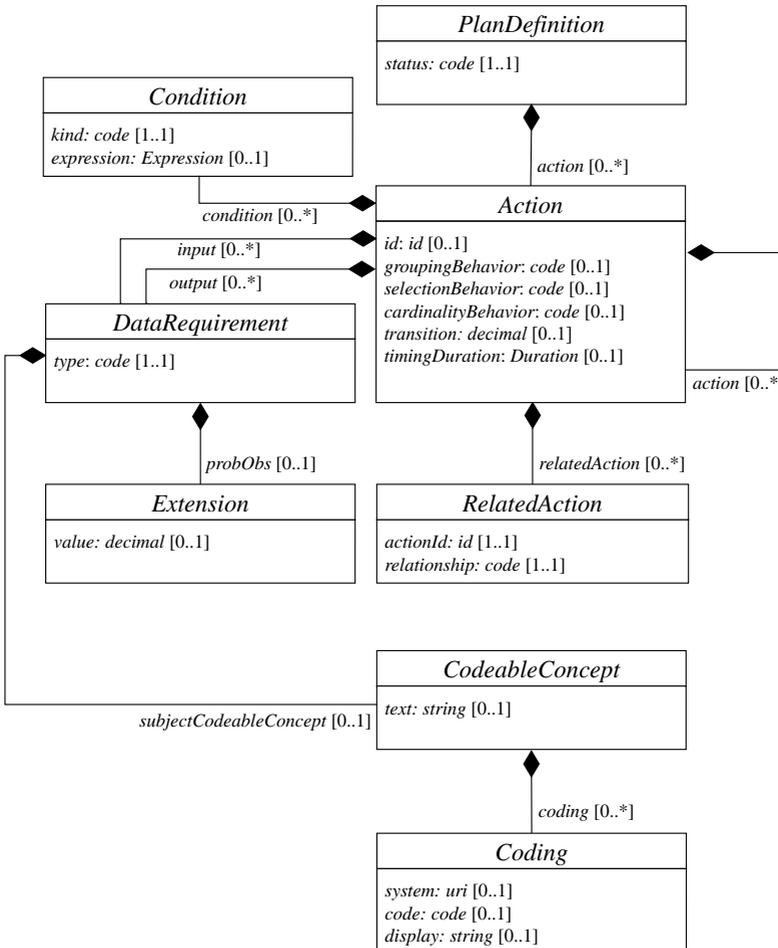


Abbildung 2.10: Die Abbildung zeigt Zusammenhänge zwischen den Klassen *PlanDefinition*, *Action* und *RelatedAction* in einem Klassendiagramm. Eine *PlanDefinition* besteht aus beliebig vielen *Actions*. Umgekehrt besteht also eine Teil-Ganze-Beziehung (schwarze Raute), wobei das Ganze die *PlanDefinition* darstellt und die Teile die einzelnen *Actions*. Eine *Action* kann aus beliebig vielen weiteren *Actions* bestehen. Zusätzlich kann jede *Action* zu beliebig vielen anderen *Actions* in Beziehung stehen sowie beliebig viele Ein- und Ausgaben vom Typ *DataRequirement* aufweisen. Letzterer wurde um einen optionalen Dezimalwert *probObs* erweitert und steht mit *CodeableConcept* und *Coding* in Verbindung.

Um darüber hinaus Wahrscheinlichkeitswerte mit einem *DataRequirement* assoziieren zu können, wurde eine Erweiterung (*Extension*) eingeführt, die einen solchen Wert als Dezimalzahl (*decimal*) aufnimmt¹. Vgl. hierzu Abbildung 2.10 mittig, links.

Gegenstand (engl. *subject*) des *DataRequirements* ist ein sogenanntes *CodeableConcept*. Referenziert wird letzteres über den Namen *subjectCodeableConcept*. Dabei handelt es sich um einen Datentyp, der aus einem beschreibenden Freitext (*text*), sowie einem oder mehreren *Codings* besteht.

Ein *Coding* bezieht sich seinerseits auf ein Codesystem (*system*), welches eine Menge möglicher *codes* definiert. Beispiele für ein solches Codesystem sind medizinische Nomenklaturen wie LOINC und SNOMED. Durch einzelne Codes sind hier beispielsweise medizinische Werkzeuge, Untersuchungswerte und Diagnosen referenzierbar. Dies erleichtert den Austausch des gemeinsam mit dem Experten formalisierten Wissens durch eine eindeutige und etablierte Begriffszuordnung.

Für einen optionalen (Daten-)input sowie einen (Daten-)output einer *Action* können also mittels *CodeableConcept* beliebig viele solcher *Coding*-Instanzen hinterlegt werden, für die jeweils das System (*system*), der Code (*code*) sowie ein beschreibender Text (*display*) angegeben werden können.

Abbildung 2.11 zeigt ein Objektdiagramm, bei dem ein *DataRequirement* als Output einer Aktion spezifiziert wurde. Das *DataRequirement* ist vom Typ *Observation* und trägt den Wahrscheinlichkeitswert *probObs.value = 0.8*.

Darüber hinaus ist dem *DataRequirement* ein *subjectCodeableConcept* mit dem beschreibenden Freitext *text=Trocar* zugewiesen. Für dieses *subjectCodeableConcept* wurde weiterhin ein *coding* hinterlegt. Das *coding* ist aus dem System SNOMED CT (*system=SNOMED CT*), trägt den Code 118418003 (*code=118418003*) und verfügt über die Bezeichnung Trocar device (*display=Trocar device*).

¹ Hinweis: Die Erweiterung wurde zur probabilistischen Modellierung von Arbeitsabläufen eigen erstellt, validiert und den FHIR-Interessensgruppen unter dem folgenden Link zur Verfügung gestellt: [https://simplifier.net/probabilisticdm/probobservation/\\$validate](https://simplifier.net/probabilisticdm/probobservation/$validate).

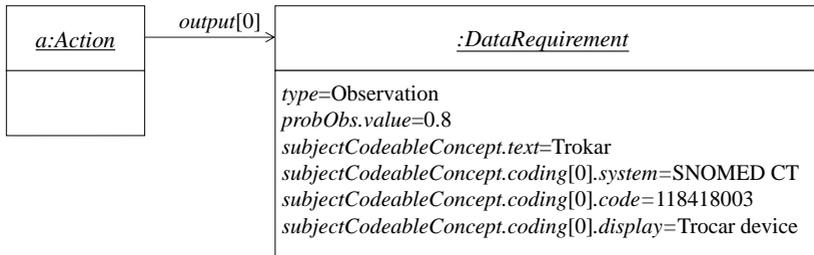


Abbildung 2.11: Die Abbildung zeigt ein Objektdiagramm mit einer Aktion *a* der Klasse *Action*, der als *output[0]* das Objekt der Klasse *Datarequirement* zugewiesen wurde. Hinweis: Aus Gründen der Übersichtlichkeit werden alle weiteren Objekte als Attribute in *output[0]* dargestellt und nicht separat mit Kante und Objektkasten visualisiert. Das Objekt *output[0]* trägt den Typ *Observation* (*type=Observation*). Weiterhin weist das Objekt einen Wahrscheinlichkeitswert von 0.8 auf, der wiederum aus dem Objekt *probObs* der Klasse *Extension* stammt und dort unter dem Attributnamen *value* bekannt ist. Darüber hinaus hat das *Datarequirement output[0]* ein *subjectCodeableConcept* mit dem beschreibenden Attribut *text=Trokar*. Für dieses *subjectCodeableConcept* wurde ein bestimmtes *coding* aus SNOMED CT hinterlegt (*system=SNOMED CT*). Der Code lautet 118418003 (*code=118418003*) und wird für den Menschen lesbar mit der Bezeichnung *Trocar device* angegeben (*display=Trocar device*).

In ähnlicher Weise lassen sich einer Aktion Bedingungen (engl. *conditions*) und eine Dauer (engl. *duration*) zuweisen.

Abbildung 2.12 zeigt wie für eine Aktion eine Bedingung *condition[0]* der Klasse *Condition* festgelegt ist. Die Art (engl. *kind*) der Bedingung betrifft die Anwendbarkeit der entsprechenden Aktion (*kind=applicability*), sprich: wann diese Aktion ausgeführt werden soll. Hierfür ist in der Bedingung wiederum ein Ausdruck (engl. *expression*) hinterlegt. Für diesen kann die Sprache (engl. *language*) und der eigentliche Ausdruck angegeben werden. Im vorliegenden Falle ist *expression.language=text/cql.identifier* und *expression.expression=x>0*

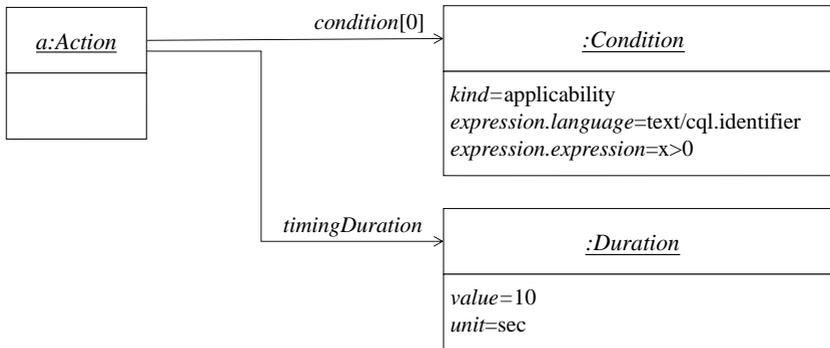


Abbildung 2.12: Die Abbildung zeigt ein Objektdiagramm mit Aktion *a* der Klasse *Action* sowie zwei zugeordnete Objekte. Das Objekt im oberen Teil der Abbildung ist aus der Klasse *Condition* (engl. Bedingung) und beherbergt deren Art (engl. kind) also auch einen zugeordneten Ausdruck (engl. expression). Für den Ausdruck ist die Sprache (engl. language) als auch der Ausdruck selbst hinterlegt. Die konkreten Ausprägungen der Attribute lassen sich wie schon gewohnt (vgl. Beschreibung Abbildung 2.11) dem Objektdiagramm entnehmen.

gewählt worden. Zusätzlich weist die gezeigte Aktion ein Dauer (engl Duration) auf¹. Sie beträgt in diesem Falle 10 Sekunden.

¹ Hinweis: Die in Abbildung 2.12 gezeigten Eigenschaften können unabhängig voneinander gesetzt werden und wurden nur zur Verdichtung von Text und Abbildungen in ein gemeinsames Schaubild zusammengefasst.

3 Prozessmodelle

Prozessmodelle und weitere Begrifflichkeiten werden in Kapitel 3.1 besprochen. In Kapitel 3.2 und 3.3 werden die Grundlagen der Petri-Netze und UML Aktivitäten dargestellt. Diese Grundlagen sind die Basis für weitergehende Betrachtungen in den nachfolgenden Teilen der vorliegenden Arbeit.

3.1 Begrifflichkeiten

Die Begriffe Prozess, Workflow, und Arbeitsablauf werden, auch in der medizinischen Domäne, unterschiedlich verwandt. Es findet sich also in der Literatur keine allgemeingültige Definition.

3.1.1 Prozess

Ein Geschäftsprozess ist gemäß [Wor99] folgendermaßen definiert¹:

„Eine Menge von einem oder mehreren verknüpften Abläufen oder Aktivitäten, die gemeinsam ein Geschäftsziel oder Strategieziel realisieren, normalerweise im Kontext einer Organisationsstruktur, die funktionale Rollen und Beziehungen definiert.“

Eine weitere Definition eines Prozesses für den Bereich der Medizin findet sich in [Neu12]. Hier wird der Begriff eines chirurgischen Prozesses in Anlehnung

¹ Hinweis: Deutsche Übersetzung der originalen Definition: A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships.

an die Definition nach [Wor99] beschrieben. Wobei das Geschäftsziel ersetzt wird durch ein sogenanntes chirurgisches Ziel. Allgemeiner zu [Wor99] wird durch [Vos96] der Begriff Prozess definiert als:

„Inhaltlich abgeschlossene, zeitlich und sachlogische Abfolge der Funktionen, die zur Bearbeitung eines betriebswirtschaftlich relevanten Objektes notwendig ist.“

Interessanterweise findet sich eine ähnliche Definition im Bereich der Medizin, hier allerdings für den Begriff des Workflows [Har19]:

„Ein Workflow ist eine inhaltlich abgeschlossene, zeitlich und sachlogisch zusammenhängende Kette von Funktionen oder Diensten, die zur Bearbeitung einer betriebswirtschaftlich relevanten Entität notwendig sind [...]“

Diese Definition deckt sich wiederum mit der nicht-medizinischen Sichtweise aus [Müh12].

3.1.2 Workflow

Nach [Aal13] liegt der Fokus bei der Workflowmodellierung bei der Automatisierung von Teilen eines (Geschäfts-)Prozesses. Workflows seien somit technisch motiviert und beschrieben im Gegensatz zu Prozessen weniger das Zusammenwirken von Personen oder den (realweltlichen) Ablauf der Geschehnisse. Diese Sichtweise wird weitestgehend gestützt von [Wor99, Wes12]. Diese definieren den Begriff Workflow wie folgt¹:

„Die ganze oder teilweise Automatisierung eines Geschäftsprozesses, während der Dokumente Informationen oder Aufgaben gemäß einer Menge von Prozessregeln von einem Beteiligten an einen anderen weitergereicht werden.“

In diesem Kontext wird der Begriff Workflow teils mit dem deutschen Begriff der Ablaufsteuerung und nicht mit Begriff Arbeitsablauf übersetzt [Kre99]. Was im Hinblick auf die Definition und der Betonung einer Automatisierung

¹ Hinweis: Deutsche Übersetzung der originalen Definition: The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules.

durchaus als sinnvoll erscheint. Hier bettet sich auch die Definition eines Workflows nach [Dum13] ein, der den Begriff Workflow gleichsetzt mit dem Begriff des automatisierten Geschäftsprozess und dabei die weitestgehend bedeutungsgleiche Definition nach [Wor99] widerspiegelt¹.

3.1.3 Arbeitsablauf

Der deutsche Begriff Arbeitsablauf (engl. Work Process [ISO16]) findet im Bereich der Arbeitswissenschaften Anwendung. Beispielsweise ist ein Arbeitsablauf nach DIN EN ISO 6385 [DIN16] definiert als eine „räumliche und zeitliche Abfolge des Zusammenwirkens von Arbeitenden, Arbeitsmitteln, Materialien Energie und Information innerhalb eines Arbeitssystems“. Wobei vereinfachend² gilt [DIN16]:

- Arbeitende sind Personen innerhalb des Arbeitssystems, die ein oder mehrere Tätigkeiten zur Erreichung eines Ziels durchführen
- Arbeitsmittel sind Werkzeuge [...] und andere im Arbeitssystem benutzte Komponenten
- Arbeitssystem ist ein System, welches das Zusammenwirken eines einzelnen oder mehrerer Arbeitender mit den Arbeitsmitteln umfasst [...] um die Funktion des Systems [...] unter den durch die Arbeitsaufgaben vorgegebenen Bedingungen zu erfüllen

Eine weitestgehend bedeutungsgleiche Definition findet sich in [REF15]. Diese Definitionen beinhalten nicht den Aspekt der Automatisierung und lassen sich somit der häufig verwandten englischen Begriff des Workflows nur unzureichend zur Deckung bringen. Vielmehr weisen die Definition stärkere Ähnlichkeiten zum Begriff des Prozesses auf, was angesichts der englischen

¹ Hinweis: Deutsche Übersetzung der originalen Literaturstelle: In this context, we will refer to an automated business process, also known as workflow: a process that is automated in whole or in part by a software system, which passes information from one participant to another for action, according to the temporal and logical dependencies set in the underlying process model.

² Hinweis: Eine ausführliche Auflistung weiterer Begrifflichkeiten und deren Zusammenhänge findet sich in [DIN16].

Übersetzung des Arbeitsablaufs als sogenannten Work Process [ISO16] sinnig erscheint. Die Begrifflichkeiten der Normen [DIN16, ISO16] betten sich dabei kompatibel zum Bereich der Gebrauchstauglichkeit [DIN99] und verwandter Normen ein, vgl. [Phi15a].

3.1.4 Definitionen

In dieser Ausarbeitung wird der Begriff Arbeitsablauf für die arbeitswissenschaftliche Betrachtung medizinischer Prozesse im Hinblick auf die Quellen [DIN16, ISO16, REF15] genutzt; falls notwendig wird die englische Übersetzung Work Process verwendet. Weiterhin wird in Anlehnung an [Wor99, Neu12] definiert:

Definition 3.1. Ein medizinischer Prozess ist eine Menge von einem oder mehreren verknüpften Abläufen oder Aktivitäten, die gemeinsam ein medizinisches Ziel realisieren, üblicherweise im Kontext einer Organisationsstruktur, die funktionale Rollen und Beziehungen definiert.

Der Begriff des Workflows wird in Bezug zur Automatisierung von Teilen eines medizinischen Prozesses verwendet und als Ablaufsteuerung (engl. Workflow) bezeichnet.

Weitergehend werden in Anlehnung an [Wor99, Neu09, Dum13] drei Ebenen der Begrifflichkeiten definiert (vgl. Tabelle 3.1).

Der Begriff des medizinischen Prozesses bezieht sich auf die (Real-)Welt. Daher wird in dieser Ausarbeitung die Durchführung eines tatsächlichen medizinischen Prozesses als medizinischen Prozess bezeichnet (in Anlehnung an [Neu12]). Unter Berücksichtigung der Einschränkungen von Modellen, z.B. durch Verkürzung [Fri06, Sta73], wird in dieser Arbeit definiert:

Definition 3.2. Ein medizinisches Prozessmodell ist ein Abbild eines medizinischen Prozesses, das einen vordefinierten Interessensausschnitt in einer formalen oder semi-formalen Repräsentation widerspiegelt.

Tabelle 3.1: Der Begriff medizinischer Prozess in Bezug zu den drei Sichten: Welt, Modell und Instanz. Ein (real-)weltlicher Prozess wird durch Modellbildung in ein Modell überführt. Je nach Zweck der Modellbildung kann dies mit mehr oder weniger Verkürzungen einhergehen [Fri06]. Wird ein Modell ausgeführt, also eine konkrete Instanz erstellt, so spricht man auch von Instanziierung.

Sicht	Begriff
Welt	med. Prozess
Modell	med. Prozessmodell, auch: Prozessdefinition, Prozessschema, Prozess
Instanz	med. Prozessinstanz auch: Prozessmodellinstanz, Fall (engl. Case), Szenario, Prozess

Weiterhin gilt für die Instanziierung eines medizinischen Prozessmodells in Anlehnung an [Wor99]:

Definition 3.3. Eine medizinische Prozessinstanz repräsentiert eine einzelne Ausführung eines Prozessmodells.

Als Methapher zur Unterteilung nach Tabelle 3.1 kann ein reales Wegenetz herangezogen werden. Das Modell des Wegenetzes kann beispielsweise mittels eines Graphen mögliche Fahrwege repräsentieren. Die Instanz des Modells ist dann ein konkreter Fahrweg innerhalb des Wegenetzes - wobei der Raum der Möglichkeiten zur Wegewahl durch das Modell des Wegenetzes vorgegeben wird. Aus diesem Grunde wird teils der Begriff Schema statt Modell verwendet¹.

Zur Modellbildung, in Zusammenhang mit Prozessen aus dem geschäftlichen Umfeld auch Geschäftsprozessmodellierung (engl. Business Process Modeling) (BPM) genannt [Wes12], sind gerichtete Graphen das Mittel der Wahl.

¹ Hinweis: Ähnliche Verdeutlichungen sind die Beziehung zwischen Klasse und Objekt in der Objektorientierung sowie Entitätstyp und Entität bei der Datenmodellierung

Als Stand der Technik gelten in diesem Zusammenhang beispielsweise Geschäftsprozessmodell und -notation (engl. Business Process Model and Notation) (BPMN), Ereignisgesteuerte Prozessketten (EPK), Petri-Netze, UML Aktivitätsdiagramme und Prozessbäume [Aal13, Aal16]. Die drei Letztgenannten werden in den folgenden Kapiteln dargestellt.

3.2 Petri-Netze

Der heute unter dem Namen Petri-Netz bekannte Formalismus entstand im Jahre 1962 [Pet67]. Damals hatte Carl Adam Petri in seiner Dissertation [Pet62] den Grundstein für die Verallgemeinerung des klassischen endlichen Automaten gelegt. Durch diese Verallgemeinerung wurde es möglich verteilte Systeme¹ mit unterschiedlichen Teilzuständen zu modellieren, bei denen Aktionen nebenläufig² abgearbeitet werden konnten [Rei13].

3.2.1 Notationselemente

Die Graphendarstellung eines klassischen Petri-Netzes umfasst zwei verschiedene Elemente: Knoten und Kanten. Bei Knoten lassen sich Plätze und Transitionen unterscheiden (vgl. Tabelle 3.2). Dabei repräsentieren Plätze passive Speicherkomponenten, während Transitionen aktiv den Zustand des Systems verändern und damit Aktionen widerspiegeln.

Die gerichteten Kanten eines Petri-Netzes werden als Pfeile symbolisiert. Gelegentlich wird ein einziger Doppelpfeil als Kurzschreibweise für zwei Pfeile in entgegengesetzter Richtung verwendet. In dieser Ausarbeitung wird diese Kurzschreibweise aus didaktischen Gründen nicht eingesetzt.

Formal lassen sich Knoten und Kanten über eine Netzstruktur miteinander verbinden [Rei13, Phi15c]:

¹ Hinweis: Im Falle Carl Adam Petris standen vor allem Rechnerarchitekturen im Fokus.

² Hinweis: Kausal unabhängig voneinander. Die Nebenläufigkeit ist Voraussetzung für das zeitlich überlappende Ausführen von Aktionen (Parallelität).

Tabelle 3.2: Die Menge der Knoten eines Petri-Netzes lassen sich in zwei Teilmengen aufteilen. Plätze repräsentieren Speicherkomponenten während Transitionen für Aktionen stehen, die den Zustand des Systems verändern.

Symbol	Beschreibung des Knotens
	Ein Platz ^a verkörpert eine Speicherkomponente. Im Gegensatz zur Transition, die eine aktive Systemveränderung repräsentiert, wird die Speicherkomponente als passiv bezeichnet. Das gebräuchliche Symbol ist der Kreis. Selten werden Plätze auch als Ellipsen visualisiert.
	Eine Transition verkörpert eine Zustandsänderung des modellierten Systems durch eine Aktion. Sowohl das Quadrat als auch das Rechteck sind übliche Symbole für Transitionen.

^a Hinweis: Je nach Kontext werden Plätze auch als Stellen bezeichnet.

Die Netzstruktur eines Petri-Netzes ist dabei gegeben durch

$$NST = (\mathcal{P}, \mathcal{Q}, \mathcal{F}), \quad (3.1)$$

mit \mathcal{P} der Menge der Plätze und \mathcal{Q} der Menge der Transitionen:

$$\mathcal{P} = \{p_i : i = 0, \dots, |\mathcal{P}| - 1\}, \quad (3.2)$$

$$\mathcal{Q} = \{q_i : i = 0, \dots, |\mathcal{Q}| - 1\}, \quad (3.3)$$

so dass gilt:

$$\mathcal{P} \cap \mathcal{Q} = \emptyset. \quad (3.4)$$

Die Menge der Kanten zwischen Plätzen und Transitionen (sowie umgekehrt) wird über die Flussrelation \mathcal{F} der Netzstruktur festgelegt:

$$\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{Q}) \cup (\mathcal{Q} \times \mathcal{P}). \quad (3.5)$$

Aus den Gleichungen (3.1) bis (3.5) lässt sich leicht ablesen, dass es sich bei der Netzstruktur NST sowohl um einen gerichteten als auch um einen bipartiten Graphen handelt [Phi15c].

Zur Modellierung von Zuständen und mithin eines dynamischen Verhaltens des Systems werden in Petri-Netzen sogenannte Marken (engl. Tokens) eingesetzt. Bei elementaren Systemnetzen wird nur eine Sorte von Marken genutzt – diese Marken werden üblicherweise durch schwarze Punkte symbolisiert [Rei13]. Die Verteilung der Marken auf den Plätzen wird als Markierung bezeichnet und spiegelt den Zustand des Petri-Netzes wider.

Eine sogenannte Anfangsmarkierung M_0 mitsamt der zugehörigen Netzstruktur NST wird in diesem Zusammenhang als elementares Systemnetz bezeichnet [Rei13]. Formal:

$$ESN = (NST, M_0) \quad (3.6)$$

mit Anfangsmarkierung

$$M_0 : \mathcal{P} \rightarrow \mathbb{N}. \quad (3.7)$$

Die Anfangsmarkierung repräsentiert den Anfangszustand des Netzes und gibt an, wie viele Marken $M_0(p) \in \mathbb{N}$ jeder Platz p initial trägt. Neue Markierungen M' entstehen durch schaltende Transitionen. Dabei werden Marken verbraucht bzw. erzeugt, d.h. den Plätzen hinzugefügt oder entnommen.

Grundvoraussetzung für schaltende Transitionen ist es allerdings, dass diese aktiviert sind. Eine Aktivierung für eine Transition q liegt vor wenn: $M(p) \geq 1$ für alle Plätze p im Vorbereich $\cdot q$ von q mit

$$\cdot q = \{p \in \mathcal{P} \mid (p,q) \in \mathcal{F}\}. \quad (3.8)$$

Weiterhin ist der Nachbereich von q gegeben durch:

$$q^* = \{p \in \mathcal{P} \mid (q,p) \in \mathcal{F}\}. \quad (3.9)$$

In einem Schaltschritt $M \xrightarrow{q} M'$ einer Transition q entsteht für jeden Platz p eine neue Markierung M' derart dass [Phi15c]:

$$M'(p) = \begin{cases} M(p) - 1 & \text{falls } p \in \cdot q \text{ und } p \notin q \cdot \\ M(p) + 1 & \text{falls } p \in q \cdot \text{ und } p \notin \cdot q \cdot \\ M(p) & \text{sonst} \end{cases} \quad (3.10)$$

Aus Gleichung (3.10) folgt, dass die Markierung eines Platzes beibehalten wird, wenn sich dieser sowohl im Vorbereich als auch gleichzeitig im Nachbereich einer Transition befindet. In der Graphendarstellung existiert also eine Kante von der Transition zum Platz und umgekehrt.

Zur Verdeutlichung der Zusammenhänge sind in Tabelle 3.3 vier elementare Systemnetze abgebildet.

Zeile 1 zeigt die Transition q_0 mit Platz p_0 als einziges Element des Vorbereichs $\cdot q_0$ und Platz p_1 als einziges Element des Nachbereichs $q_0 \cdot$. Die Transition q_0 ist aktiviert, da für alle Elemente des Vorbereiches, und damit für p_0 , gilt: $M(p_0) \geq 1$. Sobald q_0 schaltet, wird eine Marke von all jenen Plätzen entnommen, die sich ausschließlich im Vorbereich von q_0 befinden. Das bedeutet für den vorliegenden Fall: $M'(p_0) = M(p_0) - 1 = 0$.

Darüber hinaus wird den Plätzen, die sich nur im Nachbereich befinden (sprich: p_1) eine Marke hinzugefügt: $M'(p_1) = M(p_1) + 1 = 1$. Die Transition q_0 ist nach dem Schalten nicht mehr aktiviert, da Platz p_0 im Vorbereich der Transition q_0 nun keine Marke trägt.

Die Transition q_0 des elementaren Systemnetzes in Zeile 2 der Tabelle 3.3 ist aktiviert: alle Plätze des Vorbereichs tragen eine Marke. Beim Schalten bleibt allerdings die Marke in Platz p_0 erhalten, da sich p_0 gleichzeitig im Vor- als auch im Nachbereich der Transition q_0 befindet. Für Platz p_1 , der sich ausschließlich im Nachbereich befindet, ergibt sich die Markierung $M'(p_1) = 1$ analog zu Zeile 1. Nach dem Schalten bleibt die Transition allerdings aktiviert, da sich im Vorbereich der Transition nach wie vor eine Marke befindet.

Tabelle 3.3: Jede der vier Zeilen der Tabelle zeigt zwei Markierungen eines anderen elementaren Systemnetzes ESN – einmal bevor Transition q_0 geschaltet hat und einmal nachdem Transition q_0 geschaltet hat. Beim Schaltvorgang können Marken verbraucht und auch erzeugt werden.

Netz	Vor dem Schalten	Nach dem Schalten
a)		
b)		
c)		
d)		

Analog zu Zeile 1 ist die Transition q_0 das elementaren Systemnetzes in Zeile 3 zunächst aktiviert. Beim Schalten werden in allen Plätzen des Nachbereichs der Transition q_0 Marken erzeugt. Dadurch entstehen neue Markierungen $M'(p_1) = 1$ und $M'(p_2) = 1$. Da $M'(p_0) \not\geq 1$ ist die Transition nach dem Schaltschritt nicht mehr aktiviert.

Zeile 4 der Tabelle 3.3 zeigt ein weiteres elementares Systemnetz. Auch hier ist die Transition q_0 aktiviert, denn für die Plätze im Vorbereich der Transition gilt: $M(p_0) = 2$ und $M(p_1) = 1$ und damit sind die Markierungen aller Plätze im Vorbereich ≥ 1 . Im Schaltschritt der Transition werden Marken im Vorbereich entfernt, also $M'(p_0) = M(p_0) - 1 = 0$ und $M'(p_1) = M(p_1) - 1 = 1$. Weiterhin werden Marken im Nachbereich erzeugt: $M'(p_2) = M(p_2) + 1 = 1$. Eine Aktivierung der Transition q_0 liegt nun nicht mehr vor. Denn, da $M'(p_0) = 0$, ist $M'(p) \not\geq 1$ für alle Plätze p im Vorbereich.

Analog zu den Definition von Vor- und Nachbereichen von Transitionen (vgl. Gleichungen (3.8) und (3.9)), sei nun der Vorbereich eines Platzes p gegeben durch:

$${}^*p = \{q \in \mathcal{Q} \mid (q,p) \in \mathcal{F}\}. \quad (3.11)$$

Weiterhin sei der Nachbereich von p gegeben durch:

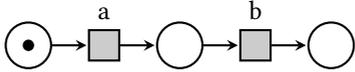
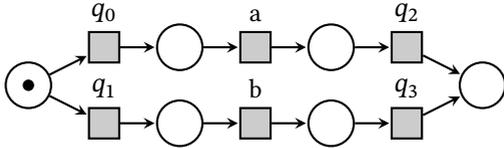
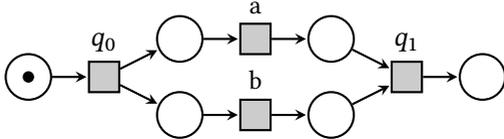
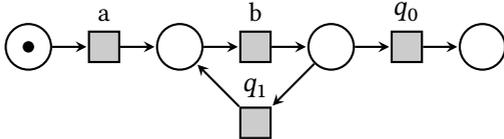
$$p^* = \{q \in \mathcal{Q} \mid (p,q) \in \mathcal{F}\}. \quad (3.12)$$

In Zeile 4 der Tabelle 3.3 gilt für den Platz p_2 beispielsweise: ${}^*p_2 = \{q_0\}$ und $p_2^* = \emptyset$. Für den Platz p_0 gilt andererseits: ${}^*p_0 = \emptyset$ und $p_0^* = \{q_0\}$.

3.2.2 Verschaltungen

Bei einem Arbeitsablauf lassen sich typische Verschaltungen unterscheiden: sequentiell, selektiv, nebenläufig und iterativ [Van04]. Tabelle 3.4 zeigt Repräsentanten solcher Verschaltungen als Petri-Netz.

Tabelle 3.4: Typische Verschaltungen im Kontext von Arbeitsabläufen – dargestellt als Petri-Netze. Die Plätze tragen aus Gründen der Übersichtlichkeit keine Namen.

Verschaltung	Beispielhafter Graph
Sequentiell	
Selektiv	
Nebenläufig	
Iterativ	

Die sequentielle Verschaltung von Transitionen (vgl. Tabelle 3.4) stellt sicher, dass Aktionen eines Arbeitsablaufs (repräsentiert durch Transitionen) nacheinander ausgeführt werden. Im Beispiel kann Transition b erst schalten, wenn zuvor Transition a geschaltet hat.

Für das explizite Modellieren einer selektiven Verschaltung werden Hilfstransitionen benötigt, die in Zeile 2 der Tabelle 3.4 mit q_0 und q_1 angegeben sind. Eine Marke wird dadurch entweder im Vorbereich der Transition a oder aber im Vorbereich der Transition b erzeugt. Dieses Teilkonstrukt wird auch als OR-Split bezeichnet. Der OR-Split bewirkt also, dass entweder die eine oder aber die andere Aktion des Arbeitsablaufs zur Ausführung kommen kann. Das Gegenstück zum OR-Split ist der sogenannte OR-Join, der durch die Transitionen q_2 und q_3 repräsentiert wird. Die Transitionen sorgen durch ihren gemeinsamen Nachbereich dafür, dass die einzelnen Stränge wieder gebündelt werden.

Um Nebenläufigkeiten zu modellieren, sind Transitionen notwendig, deren Nach- bzw. Vorbereich mehrere Plätze aufweist. In Zeile 3 der Tabelle 3.4 sorgt etwa die Transition q_0 dafür, dass jeweils eine Marke im Vorbereich der Transition a und auch der Transition b erzeugt wird. Dieses Konstrukt, auch AND-Split genannt, bewirkt also, dass sowohl a als auch b zur Ausführung gebracht werden können – und zwar in beliebiger Reihenfolge. Um beide Stränge wieder zu vereinen wird ein AND-Join verwendet, welche das Schalten weiterer Transitionen erlaubt, sobald die Transitionen beider Stränge geschaltet haben.

Für eine Iteration werden zwei Hilfstransitionen benötigt, die selektiv die Iteration fortführen oder aber beenden können. Zeile 4 der Tabelle 3.4 zeigt ein anschauliches Beispiel. Hierbei wird Transition a genau einmal ausgeführt, wohingegen Transition b beliebig oft zur Ausführung gebracht werden kann. Dabei dient selektiv Hilfstransition q_1 zum Fortführen der Iteration oder aber q_0 zum Beenden der Iteration.

3.2.3 Workflow-Netze

Sogenannte Workflow-Netze (engl. Workflow Nets) sind eine Unterklasse der Petri-Netze, welche von van der Aalst und Kollegen speziell für die Modellierung von Geschäftsprozessen definiert wurden [Aal97, Aal98].

Eine Netzstruktur $NST = (\mathcal{P}, \mathcal{Q}, \mathcal{F})$ mit $p_{\text{Quelle}} \neq p_{\text{Senke}} \in \mathcal{P}$ ist genau dann ein Workflow-Netz, wenn folgende drei Bedingungen gelten [Aal05, Zel20]:

$$\cdot p_{\text{Quelle}} = \emptyset, \quad \nexists p \in \mathcal{P} \setminus \{p_{\text{Quelle}}\} : \cdot p = \emptyset, \quad (3.13)$$

$$p_{\text{Senke}} \cdot = \emptyset, \quad \nexists p \in \mathcal{P} \setminus \{p_{\text{Senke}}\} : p \cdot = \emptyset, \quad (3.14)$$

$$\forall x \in (\mathcal{P} \cup \mathcal{Q}) : x \text{ liegt auf einem Pfad von } p_{\text{Quelle}} \text{ nach } p_{\text{Senke}}. \quad (3.15)$$

Dabei spezifizieren die ersten beiden Bedingungen (3.13) und (3.14), dass das Workflow-Netz genau einen ausgezeichneten Platz p_{Quelle} hat, der als alleinige Quelle des Netzes fungiert – weiterhin existiert genau ein ausgezeichnete Platz p_{Senke} , der als alleinige Senke des Netzes dient. Hierfür muss zunächst gelten, dass der Vorbereich des Platzes p_{Quelle} leer ist. Genauso muss der Nachbereich des Platzes p_{Senke} leer sein. Weiterhin dürfen keine alternativen Plätze innerhalb des Netzes existieren, für welche Vor- bzw. Nachbereich ebenso leer sind.

Motivation für die spezielle Struktur der Workflow-Netze mit ausgezeichneter Quelle und Senke ist, dass jeder Workflow als Automatisierung eines Geschäftsprozesses letztlich der Wertschöpfung dient und demnach einen wohldefinierten Start und ein wohldefiniertes Ende hat, zwischen denen alle Aktionen liegen [Lau20].

Weiterhin wird durch die dritte Bedingung (3.15) sichergestellt, dass das Petri-Netz stark zusammenhängend (engl. strongly connected) ist. Dies bedeutet für das Workflow-Netz, dass ein beliebiges Element aus der Menge seiner Plätze oder aber der Menge seiner Transitionen auf einem gerichteten Pfad von dessen Quelle zu dessen Senke liegen muss. Durch diese Bedingung werden sogenannte dangling tasks and/or conditions [Aal97] verhindert. Sprich: es wird sichergestellt, dass es keine Aufgaben oder Bedingungen im luftleeren Raum

gibt, die nicht zur Abarbeitung des Workflows dienen. Weiterhin wird sichergestellt, dass es sich bei dem durch das Netz repräsentierten Prozess nicht um einen Prozess handelt, der eigentlich in mehrere, vollständig voneinander unabhängige Teilprozesse unterteilt werden könnte.

Man kann zeigen, dass es sich bei den aus den drei obigen Bedingungen resultierenden Workflow-Netzen um spezielle Petri-Netze handelt, die die Eigenschaft der sogenannten Soundness [Aal96] mit sich bringen. Dabei bezeichnet man ein Petri-Netz genau dann als sound, wenn es folgende für die (Geschäfts-)Prozessmodellierung wünschenswerte Eigenschaften aufweist [Aal16, Lau20]:

- Das Netz kann terminieren: Ausgehend von jeder Markierung, die aus einer möglichen Anfangsmarkierung heraus erreichbar ist, kann eine Markierung mit nur einer Marke auf dem Endplatz erreicht werden. Sprich: ein modellierter Prozess kann also immer abgeschlossen werden.
- Das Netz terminiert richtig: Für jede Markierung, die aus einer möglichen Anfangsmarkierung heraus erreichbar ist, muss gelten: Besitzt diese Markierung eine Marke auf dem Endplatz, so müssen alle anderen Plätze leer sein. Somit dürfen also nach Abschluss des modellierten Prozesses keine Marken mehr im Netz verbleiben.
- Das Netz besitzt keine überflüssigen Transitionen: Für jede Transition muss eine aus der Anfangsmarkierung heraus erreichbare Markierung existieren, die diese Transition aktiviert. Jede Transition ist damit Teil eines Prozessschrittes, der tatsächlich im modellierten Prozesses existiert.
- Plätze können nur eine und nicht mehrere Marken tragen.

Spezielle Workflow-Netze, für die bestimmte syntaktische Anforderungen gelten und die daher konstruktionsbedingt immer die Eigenschaft der Soundness (vgl. Kapitel 3.2.3) in sich tragen, sind die sogenannten blockstrukturierten Workflow-Netze (engl. block-structured workflow-nets) [Aal16]. Zwar existieren in der Literatur unterschiedliche Definitionen [Aal16], aber

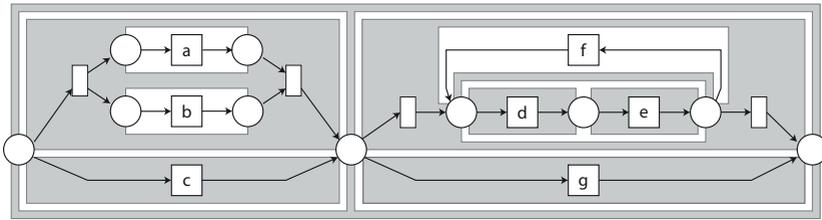


Abbildung 3.1: Die Abbildung aus [Lee14] zeigt ein Beispiel für ein blockstrukturiertes Workflow-Netz. Dieses lässt sich rekursiv in kleinere blockstrukturierte Workflow-Netze unterteilen. Letztlich besteht ein blockstrukturiertes Workflow-Netz aus mehreren aneinander gehängten und verschachtelten Blöcken. Auf der oberen Ebene sind zwei Blöcke sequentiell verschaltet. Der erste dieser beiden Blöcke lässt sich weiter aufteilen in eine selektive Verschaltung, die im oberen Ast weiter in eine nebenläufige Verschaltung aufgeteilt wird. Auch der zweite Block besteht aus einer selektiven Verschaltung. Die oberen beiden Äste werden weiter unterteilt in eine Iteration, sowie eine sequentielle Verschaltung.

üblicherweise spricht man dann von einem block-strukturierten Workflow-Netz, wenn sich dieses rekursiv anhand von Basisstrukturen erzeugen lässt [Lóp20], bzw. wenn sich ein gegebenes Workflow-Netz rekursiv in kleinere blockstrukturierte Workflow-Netze unterteilen lässt [Zel20, Lee14].

Als Resultat sind in einem blockstrukturierten Workflow-Netz alle sequentiellen, selektiven und nebenläufigen Verschaltungen sowie Iterationen als Blöcke modelliert, die einen eindeutigen Anfang und ein eindeutiges Ende besitzen [Lau20]. Das bedeutet, dass es zu jeder Verzweigung eine korrespondierende Vereinigung geben muss, beispielsweise also für jeden AND-Split ein zugehöriger AND-Join existieren muss [Aal16]. Die einzelnen Blöcke dürfen in einem blockstrukturierten Workflow-Netz verschachtelt werden, sich aber nicht überlappen [Lau20].

Abbildung 3.1 zeigt ein Beispiel eines blockstrukturierten Workflow-Netzes: Letztlich besteht das blockstrukturierte Workflow-Netz aus mehreren aneinandergehängten und/oder verschachtelten Blöcken. Auf der obersten Ebene gibt es zwei sequentiell verschaltete Blöcke, die ihrerseits wieder in Substrukturen aus selektiven, nebenläufigen und sequentiellen Verschaltungen, sowie einer Iteration zerlegt werden können.

3.3 UML Aktivitätsdiagramme

Im Jahr 1996 entstand durch die Verbindung der Arbeiten von Jim Rumbaugh, Ivar Jacobson und Grady Booch die sogenannte Unified Modeling Language (UML) in Version 0.9¹. Da in den 1990er Jahre eine Vielzahl an widersprüchlichen Modellierungsmöglichkeiten existierte, begrüßten führende Unternehmen wie Oracle, Microsoft und IBM die Bestrebungen nach einer gemeinsamen, intuitiv verständlichen Modellierungssprache [Kec21].

Diese Zeit wird teils mit dem Begriff der Notationskrise umschrieben. Neue Konzepte, wie die der Objektorientierung, konnten zu dieser Zeit noch nicht einheitlich dargestellt und ausgetauscht werden. Dies hat sich mit Einführung der UML 1.3, die letztlich als Gewinnerin aus der Notationskrise hervorging, grundlegend geändert.

Sechs Jahre nach der Zertifizierung und Freigabe der UML 1.3 durch das Standardisierungsgremium der Object Management Group (OMG) in 1999, wurden mit Version 2 grundlegende Modifikationen seitens der OMG durchgeführt. Im Zuge dieser Änderungen wurden Aktivitätsdiagramme der Semantik der Petri-Netze (siehe Kapitel 3.2, Seite 52) angepasst und orientieren sich damit am Konzept der sogenannten Marken² [OMG17]. In dieser Arbeit werden Diagramme der UML in Version 2 genutzt.

3.3.1 Notationselemente

In der aktuellen UML 2 Spezifikation [OMG17] werden 14 Diagrammtypen definiert. Jede dieser Diagrammtypen lässt sich entweder den sogenannten Struktur- oder aber den Verhaltensdiagrammen zuordnen [OMG17]. Mit ersteren lassen sich die statischen Aspekte eines Systems modellieren – z.B. die feste Aufteilung eines Systems in Klassen (Klassendiagramm). Mit Verhaltensdiagrammen kann hingegen das dynamische Verhalten eines Systems und seiner Komponenten formalisiert werden [Phi15b, Phi16b].

¹ Anmerkung: Der Kreis der Autoren wird auch als „Die drei Amigos“ bezeichnet

² Hinweis: Die Marken werden im Gegensatz zu Petri-Netzen nicht explizit dargestellt.

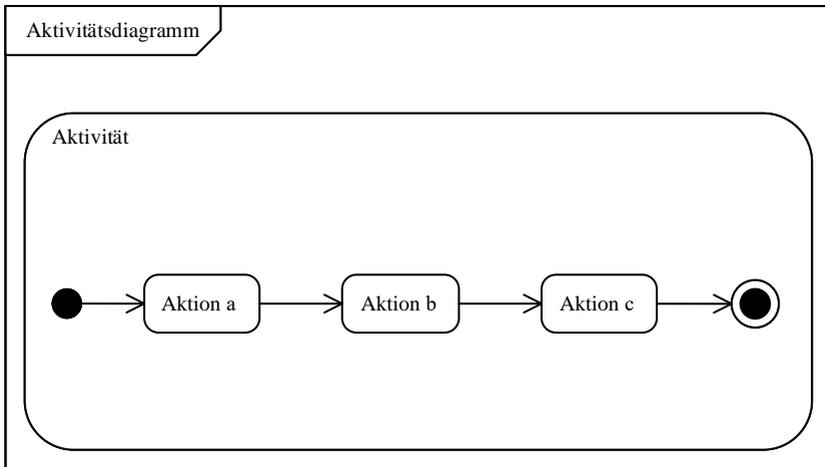


Abbildung 3.2: Zusammenhang zwischen Aktivitätsdiagramm, Aktivität, Aktion. Ein Aktivitätsdiagramm beinhaltet die Aktivität – diese legt durch die Verknüpfung von Aktionen mögliche Abläufe fest.

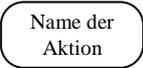
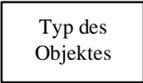
Aktivitätsdiagramme zählen zu den Verhaltensdiagrammen, geben also eine Antwort darauf, wie bestimmte Abläufe abgearbeitet werden [Rup12, Phi17d]. Dazu lassen sich in einer Aktivitäten verschiedene Notationselemente, wie z.B. Aktionen nutzen.

Abbildung 3.2 zeigt die Zusammenhänge zwischen Aktivitätsdiagramm, Aktivität und Aktionen. Das Aktivitätsdiagramm beinhaltet die Aktivität. Innerhalb einer Aktivität können Knoten, wie z.B. Aktionen, mittels gerichteter Kanten verknüpft werden, um die Menge möglicher Abläufe festzulegen.

Bei der Menge der (Aktivitäts-)knoten lassen sich drei Teilmengen unterscheiden [Kec21]:

1. Menge der ausführbaren Knoten
2. Menge der Objektknoten
3. Menge der Kontrollknoten

Tabelle 3.5: Wichtige Repräsentanten der Aktivitätsknoten der Teilmengen 1 und 2.

Notationselement	Beschreibung
	Eine Aktion repräsentiert einen Schritt innerhalb einer Aktivität. Dieser wird in der Aktivität nicht weiter zerlegt. Eine Aktion kann mit anderen Aktionen, Objektknoten oder Kontrollknoten verknüpft sein.
	Ein Objektknoten dient als Stellvertreter für Instanzen eines festgelegten Typs (z.B. bestimmte Variablen oder aber Konstanten). Er kann ebenso wie eine Aktion mit anderen Knoten verknüpft sein.

Zur Menge der ausführbaren Knoten zählen die (atomaren) Aktionen (vgl. Tabelle 3.5). Sie werden durch abgerundete Rechtecke dargestellt und verkörpern die kleinste Einheit ausführbarer Funktionalität [OMG17]– sie werden in der betreffenden Aktivität nicht weiter zerlegt.

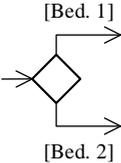
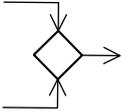
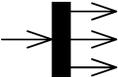
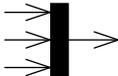
Ein weitere Teilmenge der Aktivitätsknoten ist die Menge der Objektknoten (vgl. Tabelle 3.5). Dargestellt werden Objektknoten durch ein Rechteck. Der Name des Knotens repräsentiert dabei lediglich einen bestimmten (Instanz-)Typ, ist also ein Stellvertreter für mögliche Ausprägungen der Daten in Form konkreter Instanzen¹.

Durch Verknüpfung von Objektknoten und Kanten können Datenflüsse dargestellt werden und mithin können Ein- und/oder Ausgabeparameter ausführbarer Knoten spezifiziert werden.

Die letzte Teilmenge der Aktivitätsknoten ist die Menge der Kontrollknoten (vgl. Tabelle 3.6 auf Seite 66). Dabei können für für die Modellierung alternativer Entscheidungs- und Verbindungsknoten eingesetzt werden.

¹ Hinweis: In Zusammenhang mit Entity-Relationship Modellen kann hier auch gleichbedeutend von Entitätstypen bzw. Entitäten gesprochen werden.

Tabelle 3.6: Aktivitätsknoten Teilmenge 3: Übersicht der Kontrollknoten.

Notationselement	Beschreibung
	<p>Die Entscheidung teilt Abläufe mittels Bedingungen in Alternativen auf. Symbolisiert wird dies durch eine Raute. Die Bedingungen an den abgehenden Kanten müssen vollständig und widerspruchsfrei sein. Hinweis: Teils wird statt des Begriffs der Raute die wortwörtliche Übersetzung aus dem Englischen genutzt: Diamant.</p>
	<p>Die Verbindung bündelt mittels einer Raute alternative Abläufe. Dabei genügt das Eintreffen eines Ablaufs, um über die ausgehende Kante weitere Knoten anzusteuern.</p>
	<p>Die Gabelung nutzt Synchronisationsbalken um Abläufe in mehrere Teilabläufe aufzuspalten. Dadurch können Aktionen am Ausgang der Gabelung nebenläufig abgearbeitet werden.</p>
	<p>Die Synchronisation nutzt Synchronisationsbalken um mehrere Teilabläufe zusammenzuführen. Es wird dabei auf das Eintreffen aller verbundenen Teilabläufe gewartet.</p>
	<p>Der Start (-knoten) symbolisiert den Beginn einer Aktivität. Dieser Knoten besitzt nie eine eingehende Kante.</p>
	<p>Der Ende (-knoten) markiert den Endpunkt einer Aktivität. Alle Abläufe einer Aktivität werden durch ihn beendet.</p>

Nebenläufigkeiten können hingegen mit Gabelungs- und Synchronisationsknoten umgesetzt werden. In jedem Falle muss die Aktivität einen Start- sowie Endknoten enthalten.

Die Kanten einer Aktivität sind weitere Notationselemente. Mit ihnen lassen sich einzelne Aktivitätsknoten verbinden – die gerichteten Kanten können dabei mit Bedingungen versehen werden. Es können zwei Mengen an Kanten unterschieden werden [OMG17]:

1. Menge der Kanten des Kontrollflusses
2. Menge der Kanten des Datenflusses

Gerichtete Kanten zwischen ausführbaren Knoten untereinander sowie von oder zu Kontrollknoten repräsentieren den Kontrollfluss. Ist allerdings ein Objektknoten eine Quelle oder Senke einer gerichteten Kante, so handelt es sich um einen Datenfluss [Rup12].

Der Fluss an Daten als Eingabe- bzw. Ausgabeparameter von Aktionen kann alternativ über eine Pin-Notation verdeutlicht werden [Rup12, Stö05, Phi18e]. Tabelle 3.7 stellt beide Darstellungen gegenüber. In dieser Ausarbeitung findet die Pin-Notation Anwendung.

Tabelle 3.7: Für eine kompakte Darstellung von Datenflüssen, im Besonderen zur Hervorhebung von Eingabe- und Ausgabeparametern, kann die Pin-Notation verwendet werden. In Anlehnung an [Phi18e, Phi19a].

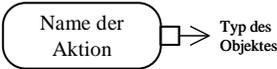
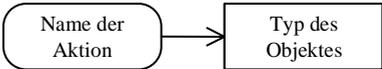
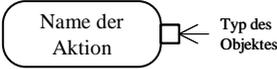
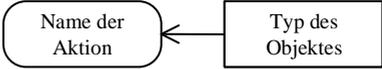
Pin Notation	Alternative Schreibweise
	
	

Tabelle 3.8: Typische Verschaltungen von Aktivitätsknoten in Anlehnung an [Phi15b, Phi16a, Phi16b, Phi17c, Phi17d]. Bei einer sequentiellen Verschaltungen werden die Aktionen a und b nacheinander ausgeführt. Im selektiven Fall wird entweder Aktion a oder aber Aktion b ausgeführt, während im nebenläufigen Falle die Aktionen a und b in beliebiger Reihenfolge ausgeführt werden können ^a.

Verschaltung	Beispielhafter Graph
Sequentiell	
Selektiv	
Nebenläufig	
Iterativ	

^a Hinweis: Einzelne Aktivitätsknoten sind in den Tabellen 3.5 und 3.6 ab Seite 65 gelistet.

3.3.2 Verschaltungen

Tabelle 3.8 zeigt in der ersten Zeile eine sequentielle Verschaltung von Aktionen, d.h. im beispielhaften Graphen wird zunächst die Aktion a und dann die Aktion b abgearbeitet.

Die zweite Zeile der Tabelle 3.8 zeigt eine selektive Verschaltung. Hier wird also in Abhängigkeit einer Entscheidung bzw. damit verknüpften Bedingungen entweder Aktion a oder aber Aktion b abgearbeitet und schließlich die Kanten über eine Verbindung wieder gebündelt.

Die dritte Zeile der Tabelle 3.8 stellt eine nebenläufige Verschaltung dar. Das bedeutet, dass die Aktionen a und b in beliebiger Reihenfolge ausgeführt werden können. Dazu wird eine Gabelung zur Aufspaltung der Kanten genutzt, die nachfolgenden Kanten werden dann durch eine Synchronisation wieder zusammengeführt.

Die letzte Zeile der Tabelle 3.8 zeigt eine iterative Verschaltung. Die Aktion a kann dabei mehrfach ausgeführt werden. Dazu wird der Aktion a eine Entscheidung nachgelagert, mit der die Iteration fortgesetzt oder beendet werden kann. Die rückwärtige Kante wird über eine Verbindung mit der Kante des Startknotens vereint.

3.3.3 FHIR Repräsentationen

Die dargestellten Verschaltungen lassen sich, wie bereits in Kapitel 2.5.3 angedeutet, in FHIR repräsentieren. Für die sequentielle Verknüpfung zweier Aktionen a und b aus Zeile 1 der Tabelle 3.8 ergibt sich die Darstellung in Abbildung 3.3. Dabei werden die Knoten der Sequenz durch die anonyme Aktion *:Action* gruppiert (*groupingBehavior=logical-group*). Alle Aktionen der Gruppe sollen ausgeführt werden (*selectionBehavior=all*).

In Abbildung 3.3 sind das die Aktion a der Klasse *Action* (*a:Action*) und die Aktion b der Klasse *Action* (*b:Action*). Darüber hinaus soll die Gruppe einmalig zur Ausführung kommen (*cardinalityBehavior=single*). Allerdings sollen die Aktionen nicht nebenläufig voneinander zum Zuge kommen, sondern die

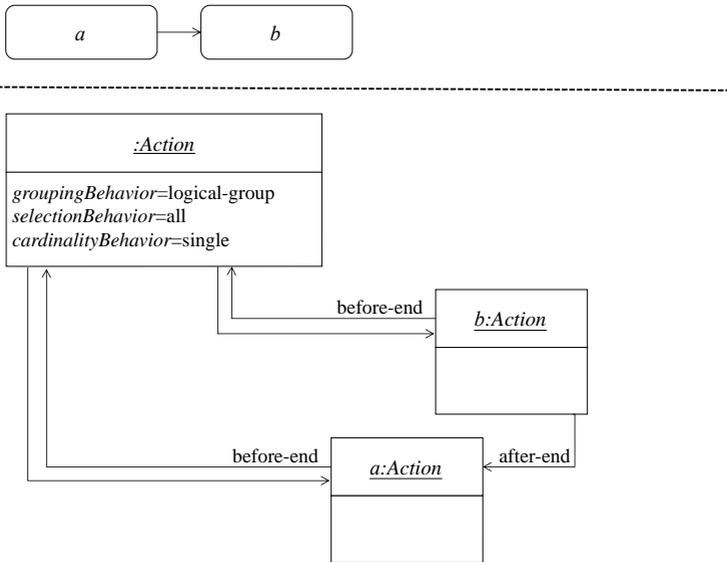


Abbildung 3.3: Die Abbildung zeigt im oberen Teil die sequentielle Verschaltung zweier Aktionen *a* und *b* als Ausschnitt aus einer UML Aktivität. Im unteren Bereich ist die entsprechende FHIR-Repräsentation verdeutlicht. Hierbei werden *a* und *b* ebenso sequentiell ausgeführt – d. h. zunächst wird Aktion *a* und dann Aktion *b* ausgeführt. Ein Knoten des Typs *Action* bündelt die Sequenz (*groupingBehavior=logical-group*) über das Attribut *selectionBehavior=all*. Die verbundenen Kindknoten *a* und *b*, von denen alle ausgeführt werden können, müssen beendet sein (*groupingBehavior=before-end*) bevor die Elternaktion endet und mit- hin die Ausführung andere Aktion außerhalb der Gruppe beginnen kann. Eine mehrfache Ausführung des Elternknotens ist ausgeschlossen (*cardinalityBehavior=single*)

Aktion *b* soll erst nach Beendigung der Aktion *a* ausgeführt werden können. Dazu wird der Aktion *b* die *relatedAction* *a* mit *relationship=after-end* zugewiesen. Weiterhin sollen alle Aktionen beendet sein, ehe weitere Aktionen außerhalb der Gruppe angestoßen werden. Dazu werden sowohl der Aktion *a* also auch der Aktion *b* die anonyme Aktion als *relatedAction* mit *relationship=before-end* zugewiesen.

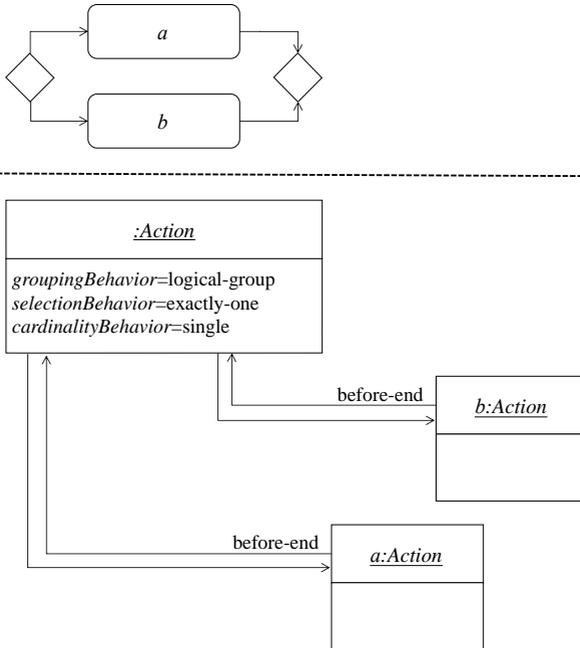


Abbildung 3.4: Die Abbildung zeigt im oberen Teil einen Ausschnitt aus einer UML Aktivität bei dem zwei Aktionen *a* und *b* selektiv verschaltet sind. Im unteren Teil ist die entsprechende FHIR-Repräsentation dargestellt. Im Unterschied zu Abbildung 3.3 weist die gruppierende, anonyme Aktion das Attribut `selectionBehavior=exactly-one` auf. Weiterhin entfällt bei der selektiven Verschaltung die sequentielle Abhängigkeit zwischen Aktion *a* und *b* (after-end).

In Abbildung 3.4 ist die selektive Verschaltung zweier Aktionen *a* und *b* dargestellt. Einmal als Ausschnitt einer UML Aktivität (oben) und einmal als FHIR-Repräsentation (unten). Ebenso wie im selektiven Falle kommt eine anonyme, gruppierende Aktion zum Einsatz. Im Unterschied zu Abbildung 3.3 weist die Gruppe allerdings ein `selectionBehavior=exactly-one` auf. Sprich: Entweder Aktion *a* oder aber Aktion *b* kann ausgeführt werden. Weiterhin bestehen zwischen den beiden Aktionen keine Abhängigkeiten, sodass solcherlei Beziehungen in der Abbildung nicht zu sehen sind.

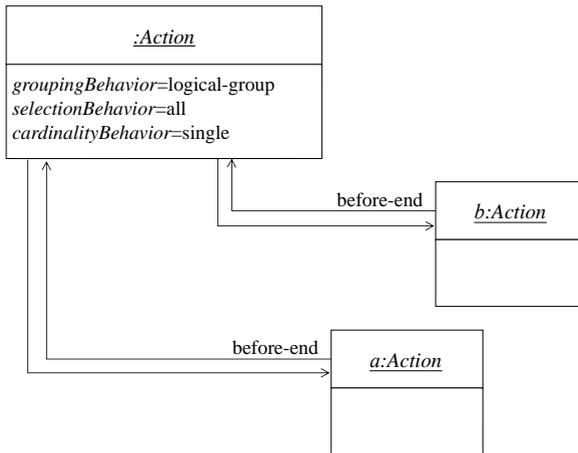
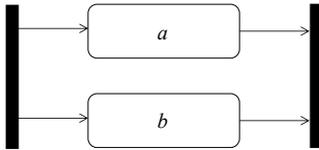


Abbildung 3.5: Die Abbildung zeigt im oberen Teil einen Ausschnitt aus einer UML Aktivität bei dem zwei Aktionen *a* und *b* nebenläufig verschaltet wurden. Im unteren Teil ist die FHIR-Repräsentation zu sehen. Im Unterschied zu Abbildung 3.4 weist die anonyme Aktion das Attribut `selectionBehavior=all` auf. Dadurch können beide Aktionen unabhängig voneinander ausgeführt werden. Allerdings: Beide Aktionen müssen beendet sein, damit die Elternaktion enden kann.

In Abbildung 3.5 findet sich die nebenläufige Verschaltung von Aktionen. Dabei wird für die gruppierende, anonyme Aktion `:Action` das Attribut `selectionBehavior=all` gesetzt. Sprich: Beide Aktionen *a* und *b* sollen ausgeführt werden. Zusätzlich können beide Aktionen nebenläufig zum Zuge kommen, da keine Abhängigkeiten zwischen beiden Aktionen gesetzt sind. Allerdings müssen beide Aktionen beendet worden sein, bevor die Elternaktion enden kann (before-end).

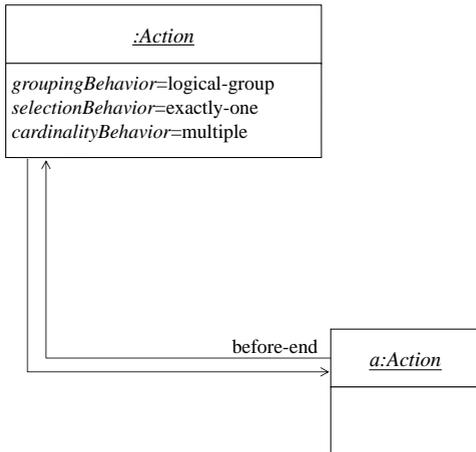
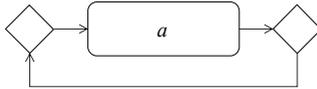


Abbildung 3.6: Die Abbildung zeigt die zeitliche Abhängigkeit zweier Aktionen *a* und *b*. Oben dargestellt als Ausschnitt eines UML Aktivitätsdiagramms und unten als Ausschnitt eines FHIR Plans. Unten zeigt die gerichtete Kante dem eigentlichen Kontrollfluss eines UML Aktivitätsdiagramms entgegen: Sobald Aktion *a* geendet hat, kann Aktion *b* ausgeführt werden (after-end).

In Abbildung 3.6 ist schließlich die iterative Verschaltung von Aktionen abgebildet (vgl. Tabelle 3.8 auf Seite 68). Die Aktion *a* wird dabei einer gruppierenden, anonymen Aktion zugeordnet (*groupingBehavior=logical-group*). Zusätzlich wird *cardinalityBehavior=multiple* gesetzt. Dadurch können Kindaktionen (in diesem Falle *a*) mehrfach ausgeführt werden.

3.4 Prozessbäume

Prozessbäume (engl. Process Trees) sind baumförmige Prozessmodelle, die als weitere Repräsentation für die in Kapitel 3.2.3 beschriebenen Workflow-Netze in [Aal11] eingeführt wurden. Da sie neben der graphischen Darstellung als Baum auch über eine textuelle Schreibweise verfügen, werden sie oft alternativ als kompakte Darstellung von Workflow-Netzen verwendet [Lee13].

3.4.1 Aufbau und Notationselemente

Ein Prozessbaum $T = (\mathcal{X}, \mathcal{Y})$ ist zunächst einmal ein gerichteter, azyklischer Graph (DAG) mit Knotenmenge \mathcal{X} und Kantenmenge \mathcal{Y} . Seine Knoten lassen sich unterteilen in:

- innere Knoten (Verzweigungsknoten mit Ausgangsgrad > 1),
- Blattknoten (Endknoten mit Ausgangsgrad 0).

Der Prozessbaum ist ein Wurzelbaum und besitzt einen speziellen inneren Knoten $r \in \mathcal{X}$, den sogenannten Wurzelknoten (vgl. Abbildung 3.7). Dieser besitzt als einziger Knoten des Baumes den Eingangsgrad 0. In einem Wurzelbaum gilt, dass gerichtete Wege¹ vom Wurzelknoten r zu jedem anderen Knoten des Baumes existieren [Hoc10]. D.h.:

$$\forall v \in \mathcal{X} \exists w_T^{r \rightarrow v} \Leftrightarrow T \text{ ist ein Wurzelbaum mit Wurzel } r.$$

¹ Hinweis: Der Begriff eines gerichteten Weges innerhalb eines Graphens wird in der Literatur nicht einheitlich verwendet. Einige Autoren bezeichnen mit dem Begriff Weg eine Folge von Knoten, z.B. [Die10] (dabei müssen bei einem gerichteten Weg die aufeinander folgenden Knoten über gerichtete Kanten des Graphen verbunden sein). Andere Autoren bezeichnen mit dem Begriff Weg dagegen eine Folge, in der sich Knoten und Kanten des Graphen abwechseln, z.B. [Tit19] (dabei gilt für gerichtete Graphen, dass kein Knoten und keine Kante mehr als einmal auftreten). Man spricht in diesem Zusammenhang auch von einem Kantenzug. Für die gegebene Beschreibung des Prozessbaums können tatsächlich beide Deutungen alternativ verwendet werden, ohne dass dies eine Auswirkung auf die Definition des Prozessbaums oder dessen Komponenten hat. In dieser Arbeit wird der Weg innerhalb eines gerichteten Graphen G als Knotenfolge definiert und der Weg vom Knoten u bis zum Knoten v durch $w_{u \rightarrow v}^G$ symbolisiert.

Weiterhin lassen sich spezielle Knotenmengen definieren:

- $Pa(v) = \{u \in \mathcal{X} \mid \exists (u, v) \in \mathcal{Y}\}$: Menge der Knoten, von denen eine gerichtete Kante nach v führt (d.h. der Elternknoten von v),
- $Desc(v) = \{u \in \mathcal{X} \mid \exists w_T^{v \rightarrow u}\}$: Menge der Knoten, die auf einem von v ausgehenden Weg liegen (d.h. die Nachfahren von v),
- $Chi(v) = \{u \in \mathcal{X} \mid \exists (v, u) \in \mathcal{Y}\}$: Menge der direkten Nachfahren von v (d.h. die Kinder von v).

Die Blattknoten des Prozessbaumes stellen die einzelnen Aktionen innerhalb des modellierten Prozesses dar. Die inneren Knoten repräsentieren sogenannte Operatoren. Diese definieren, wie die Einzelaktionen und Teilbäume – bestehend aus mehreren Aktionen und/oder Operatoren – miteinander verschaltet sind. In der grafischen Darstellung werden Blattknoten bzw. Aktionen durch Rechtecke repräsentiert, wohingegen innere Knoten bzw. Operatoren durch Kreise visualisiert werden (vgl. Abbildung 3.7). Sprich: Operatoren können nur durch die inneren Knoten des Prozessbaums repräsentiert werden (nie durch die Blattknoten) – Sie haben also immer mindestens zwei Kinder. Die Kindknoten eines Operators können dabei jedoch sowohl Blattknoten als auch innere Knoten sein.

Grundsätzlich können verschiedenste Operatoren für Prozessbäume definiert werden. Gängige Basisoperatoren, die für die medizinischen Prozesse eine Rolle spielen, sind der Operator der sequentiellen Verschaltung (repräsentiert durch das Symbol \rightarrow), der Operator der selektiven Verschaltung (repräsentiert durch das Symbol \times), der Operator der nebenläufigen Verschaltung (repräsentiert durch das Symbol \wedge) und der Schleifen-Operator (repräsentiert durch das Symbol \oslash).

Tabelle 3.9 zeigt beispielhafte Prozessbäume, die die verschiedenen grundsätzlichen Verschaltungsvarianten mithilfe der Basisoperatoren darstellen, sowie jeweils die dazu korrespondierenden Workflow-Netze. In der ersten Spalte wird zunächst die textuelle Beschreibung des Prozessbaums mithilfe der Operatoren angegeben. Die Spalten zwei und drei zeigen die graphischen Repräsentationen als Prozessbaum und Workflow-Netz. Die Aktion τ beschreibt

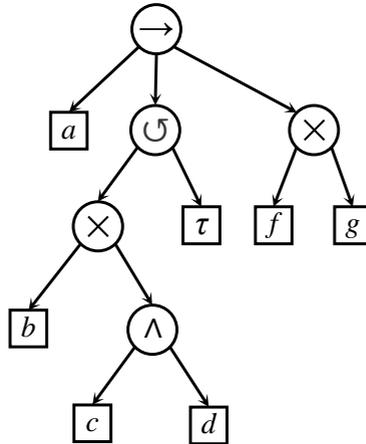


Abbildung 3.7: Beispielhafter Prozessbaum. Seine Blätter a bis g , rechteckig dargestellt, repräsentieren die Aktionen des Prozesses. Seine inneren Knoten, kreisförmig dargestellt, repräsentieren verschiedene Operatoren, die bestimmen, wie Einzelaktionen und Teilbäume miteinander verschaltet sind. Hier sind vier Basisoperatoren dargestellt: Die sequentielle Verschaltung (\rightarrow), die selektive Verschaltung (\times), die nebenläufige Verschaltung (\wedge) und die Iteration (\odot).

die sogenannte stille Aktion, eine Hilfsaktion, die für bestimmte Operatoren in Form einer Hilfstransition zum Workflow-Netz hinzugefügt werden muss.

Weiterhin fällt in Tabelle 3.9 auf, dass für zwei der vier Basisoperatoren die Reihenfolge ihrer Kindknoten eine Rolle spielt [Aal11]: Für den Operator der sequentiellen Verschaltung bestimmt die Reihenfolge seiner Kindknoten, in welcher Reihenfolge diese abgearbeitet werden (von links nach rechts).

Für den Schleifen-Operator gilt folgende Konvention, auf die auch alternative Definitionen des Schleifen-Operators in der Literatur zurückgeführt werden können [Zel20]: Der linke Kindknoten des Schleifen-Operators ist der Do-Teil der Schleife. Dieser Teil wird immer ausgeführt. Nach Ausführung des linken Kindknotens ist das Ausführen des rechten Kindknotens, des sogenannten ReDo-Teils der Schleife, optional. Wird der optionale rechte Kindknoten ausgeführt, muss anschließend nochmals der linke Kindknoten ausgeführt werden. Die Abfolge aus Do- und ReDo-Teil der Schleife kann mehrfach

Tabelle 3.9: Prozessbäume, bestehend aus den Basisoperatoren, sowie deren korrespondierende Workflow-Netze. Die Prozessbäume können sowohl in textueller Form als auch in graphischer Form dargestellt werden. Bei der Überführung von (Teil-)Bäumen mit den Operatoren \times (selektive Verschaltung), \wedge (nebenläufige Verschaltung) und \oslash (iterative Verschaltung) in ein Workflow-Netz, müssen sogenannte stille Aktionen τ als Hilfstransitionen eingefügt werden. Bei den Operatoren \rightarrow (sequentielle Verschaltung) und \oslash (iterative Verschaltung) ist die Reihenfolge der Kindknoten relevant.

Textueller Prozessbaum	Graphischer Prozessbaum	Workflow-Netz
$\rightarrow (a, b, \dots, z)$		
$\times (a, b, \dots, z)$		
$\wedge (a, b, \dots, z)$		
$\oslash (a, \tau)$		

wiederholt werden, die Ausführung der Schleife muss jedoch immer mit dem Do-Teil beendet werden. Bei den in dieser Arbeit betrachteten Prozessbäumen zur Abbildung medizinischer Prozesse wird für den ReDo-Teil jeweils die stille Aktion τ gesetzt. Durch die so definierte Verschaltung kann eine Aktion im Do-Teils einmal oder beliebig oft durchgeführt werden.

3.4.2 Formale Definition

Mithilfe der Menge der Operatoren $\{\rightarrow, \times, \wedge, \circlearrowleft\}$ und der speziellen Hilfsaktion τ können Prozessbäume auf rekursive Weise formal definiert werden als:

Definition 3.4. Gegeben sei eine Menge an Operatoren $\{\rightarrow, \times, \wedge, \circlearrowleft\}$, eine Menge an Aktionen \mathcal{A} , sowie eine stille Aktion τ .

- Dann ist a mit $a \in \mathcal{A} \cup \{\tau\}$ ein Prozessbaum.
- Seien T_0, \dots, T_{n-1} Prozessbäume und sei $\oplus \in \{\rightarrow, \times, \wedge\}$. Dann ist auch $\oplus(T_0, \dots, T_{n-1})$ ein Prozessbaum.
- Seien T_0, T_1 Prozessbäume. Dann ist auch $\circlearrowleft(T_0, T_1)$ ein Prozessbaum.

Abbildung 3.8 auf Seite 79 zeigt einen beispielhaften Prozessbaum, sowie dessen korrespondierendes Workflow-Netz. Der abgebildete Prozessbaum lässt sich mithilfe der Operatoren auch in folgender kompakter Schreibweise darstellen: $\rightarrow(a, \circlearrowleft(\times(\wedge(b, c, d), e, f), \tau), \wedge(h, k))$.

Prozessbäume sind als alternative Repräsentation von Workflow-Netzen zu sehen und lassen sich daher eins zu eins in diese überführen [Zel20].

Die umgekehrte Transformation von einem Workflow-Netz in einen Prozessbaum ist komplexer und wurde in [Zel20] betrachtet. Dabei konnte ein Algorithmus entwickelt werden, welcher für die Unterklasse der blockstrukturierten Workflow-Netze die korrekte Übersetzung in einen Prozessbaum garantiert.

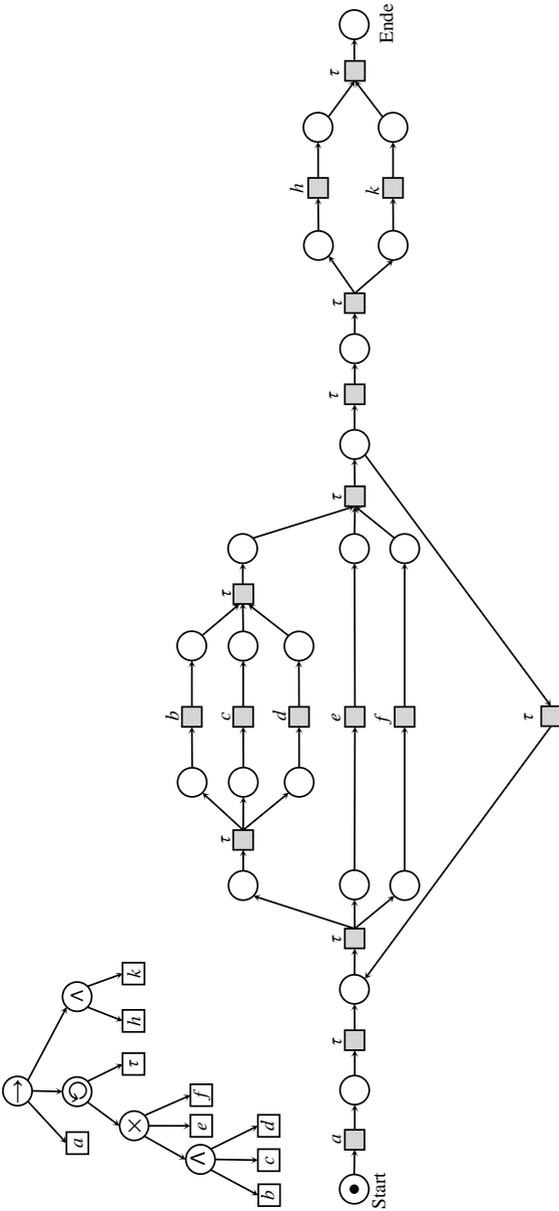


Abbildung 3.8: Die Abbildung zeigt links einen Prozessbaum und rechts dessen korrespondierendes Workflow-Netz. Auf der obersten Ebene des Prozessbaums werden die Aktion a und zwei weitere Teilbäume sequentiell verschaltet. Der rechte der auf der obersten Ebene sequentiell verschalteten Teilbäume steht für eine nebenläufige Verschaltung der Aktionen h und k . Der mittlere der auf der obersten Ebene sequentielle verschalteten Teilbäume beschreibt ein Schleifen-Konstrukt, in das eine selektive Verschaltung und eine nebenläufige Verschaltung verschachtelt sind. Dabei entspricht nach Konvention der rechte Kindknoten des Schleifen-Operators dem optionalen ReDo-Teil der Schleife. Der linke Kindknoten, in diesem Fall der Operator für die selektive Verschaltung, entspricht dem Do-Teil der Schleife, welcher mindestens ein Mal ausgeführt werden muss.

4 Transformationen zwischen Prozessbäumen und FHIR Plänen

Im folgenden Kapitel 4.1 wird das Gesamtkonzept vorgestellt, welches zur Einordnung der Transformation zwischen Prozessbäumen (vgl. Kapitel 3.4) und FHIR-Plänen (vgl. Kapitel 2.5.3) dient. Danach werden zunächst Prozessbäume in Kapitel 4.2 und nachfolgend FHIR-Pläne in Kapitel 4.3 als Ausgangsmodele beleuchtet. Anschließend folgt in Kapitel 4.4 die Transformation eines Prozessbaumes in einen FHIR-Plan, sowie die Rücktransformation in Kapitel 4.5. Schließlich wird in Kapitel 4.6 gezeigt, inwieweit die vorgestellten Transformationen die Struktur der Originalmodelle erhalten.

4.1 Gesamtkonzept

Abbildung 4.1 zeigt das Gesamtkonzept, in das sich die Beiträge dieser Arbeit einordnen lassen. Das Konzept beschreibt ein WBS, genauer gesagt ein Expertensystem. Wie in Kapitel 2 erläutert haben solcherlei Systeme eine Wissensbasis, die von der Wissensverarbeitung getrennt ist. Im vorliegenden Fall wurde, aus den in Kapitel 2.5 ausführlich dargelegten Gründen, FHIR als Wissensbasis gewählt. Die eigentliche Wissensverarbeitung erfolgt in der Problemlösungskomponente.

Grundvoraussetzung für ein Expertensystem ist der Wissenserwerb, vgl. Kapitel 2.2. Hierzu muss das vorhandene Expertenwissen formalisiert werden – im Falle des hier betrachteten Anwendungsszenarios ist dies das Wissen über medizinische Prozesse. Ziel der Formalisierung ist letztlich ein für alle

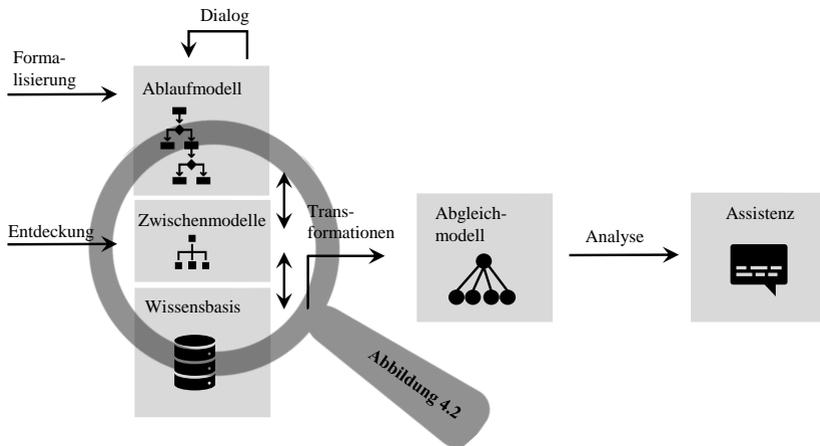


Abbildung 4.1: Die Formalisierung des menschlichen Wissens kann im Dialog der medizinischen und technischen Domänenexperten vollzogen werden. Ergebnis ist ein für alle Seiten verständliches Ablaufmodell, welches über Zwischenmodelle in die FHIR-basierte Wissensbasis gelangt. Weitere Details finden sich in *Abbildung 4.2*. Zusätzlich können Methoden zur Entdeckung von Abläufen (vgl. Process Mining) über die Zwischenmodelle eingebunden werden. Aus der Wissensbasis können wiederum Abgleichmodelle für die Bereitstellung von Assistenzfunktionen erzeugt werden (Problemlösungskomponente). Dieser Teil des Konzeptes wird in Kapitel 5.4 für eine Phasendetektion bei chirurgischen Interventionen exemplarisch dargestellt und in nachfolgenden Kapiteln bewertet.

Seiten verständliches Ablaufmodell, welches die Abfolge der einzelnen Prozessschritte des medizinischen Prozesses beschreibt und in einem Dialog zur Diskussion gestellt werden kann.

Grundsätzlich können für ein solches Ablaufmodell unterschiedliche Prozessmodelle herangezogen werden, vgl. Kapitel 3. Aufgrund ihrer Verständlichkeit auch für nicht-technische Nutzer eignen sich insbesondere grafische Prozessmodelle, wie beispielsweise Geschäftsprozessmodell und -notation (BPMN), Ereignisgesteuerte Prozessketten (EPK) oder UML Aktivitätsdiagramme [Ko09]. In dieser Arbeit werden UML Aktivitätsdiagramme, aus den in Kapitel 3.3 dargestellten positiven Eigenschaften heraus, zur Modellierung genutzt.

Die darin enthaltenen Informationen über einen medizinischen Prozess müssen letztlich für die Wissensbasis des Expertensystems verfügbar gemacht werden. Dafür müssen die UML Aktivitätsdiagramme in FHIR überführt werden, sodass die Abläufe schließlich als FHIR-Pläne hinterlegt werden können. Dies erfolgt durch drei Transformationen, siehe Abbildung 4.2.

Zunächst wird das im Expertendialog erstellte grafische Ablaufmodell in ein Workflow-Netz, d.h. in ein spezielles Petri-Netz wie in Kapitel 3.2.3 beschrieben, überführt. Dies kann mittels der in [Fav15] vorgestellten Übersetzungsregeln erfolgen. Hierbei ist die konkrete grafische Modell, d.h. ob das grafische Modell beispielsweise mittels BPMN oder aber UML Aktivitätsdiagrammen dargestellt wird, nicht von Relevanz, da jede dieser Modell zur Formalisierung des Kontrollflusses des Workflow-Graphen nutzt.

Auch eine Rücktransformation von einem Workflow-Netz in das grafische Ablaufmodell ist möglich. Voraussetzungen für die Übersetzungsregeln ist, dass es sich um ein sogenanntes free-choice Workflow-Netz handelt. In einem solchen Workflow-Netz haben je zwei beliebige Plätze entweder exakt die selbe Menge an Ausgangstransitionen, oder aber der Schnitt der Mengen der Ausgangstransitionen beider Plätze ist leer [Esp17].

Dadurch gilt für jede erreichbare Markierung im Netz folgender Sachverhalt: Falls eine Ausgangstransition eines Platzes feuern kann, können auch alle anderen Ausgangstransitionen dieses Platzes feuern – es steht also frei zu entscheiden (engl. free choice), welche Ausgangstransition letztlich feuert [Esp17]. Diese Voraussetzung ist für die zur Modellierung der medizinischen Prozesse betrachteten block-strukturierten Prozessmodelle implizit gegeben.

Ist das grafische Ablaufmodell in ein Workflow-Netz transformiert, kann dieses in einem weiteren Transformationsschritt in einen Prozessbaum überführt werden [Zel20]. Auch hier ist eine Rücktransformation möglich. Aus dem so generierten Prozessbaum wird schließlich mittels der in dieser Arbeit entwickelten Transformation ein FHIR-Plan erzeugt.

Die Repräsentation als Prozessbaum kann weiterhin als Schnittstelle zum Feld des Process Minings gesehen werden. Ziel dieses Forschungsgebiets ist, vereinfacht gesagt, das Herausschälen von implizitem Prozesswissen aus Daten,

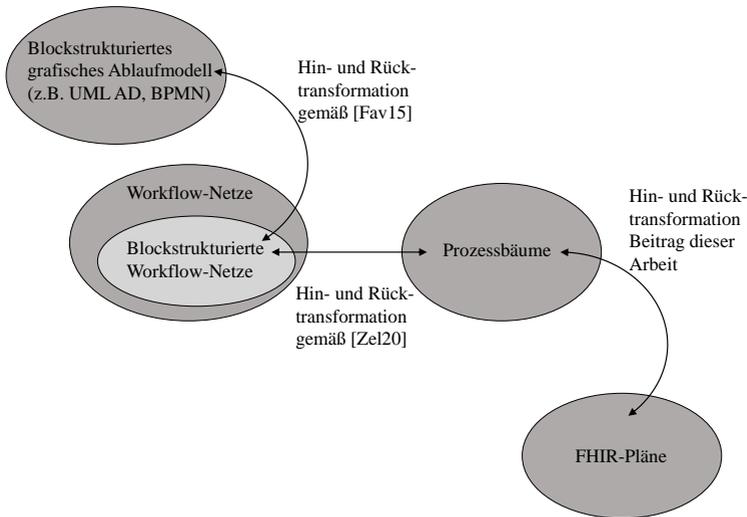


Abbildung 4.2: Darstellung der Überführung eines im Expertendialog gewonnenen Ablaufmodells in die Wissensbasis in FHIR. Zunächst erfolgt die Überführung des UML Aktivitätsdiagramms – oder eines beliebigen anderen grafischen Prozessmodells – in ein Workflow-Netz, d.h. ein spezielles Petri-Netz. Dies erfolgt anhand der in [Fav15] beschriebenen Übersetzungsregeln. Das Workflow-Netz kann anschließend als Prozessbaum repräsentiert werden [Zel20]. Die anschließende Überführung eines Prozessbaums in einen FHIR-Plan ist ein Beitrag dieser Arbeit. Für alle drei Teilschritte ist auch eine Rücktransformation möglich, wenn bestimmte Einschränkungen an die Modelle gestellt werden^a.

^a Hinweis: In der Abbildung wird aus Platzgründen der Begriff UML Aktivitätsdiagramm durch UML AD abgekürzt.

die durch ein technisches System während der wiederholten Ausführung von Prozessinstanzen generiert wird und letztlich eine Sammlung an Ablaufprotokollen des Prozesses darstellt, sogenannte Event Logs [Aal16].

Während unterschiedliche Techniken zum Entdecken von Prozessen aus Event Logs existieren, gibt es eine Klasse von induktiven Methoden, die sogenannten Inductive Mining Verfahren [Lee13], die garantieren, dass die entdeckten Prozessmodelle zu den beobachteten Abläufen passen und zudem die wünschenswerte Eigenschaft der Soundness mit sich bringen, vgl.

Kapitel 3.2.3. Dererlei Techniken basieren auf Prozessbäumen und liefern dementsprechend Prozessbäume als Ergebnis des Herausschälens zurück. Weiterhin gibt es interessante heuristische Methoden basierend auf genetischen Algorithmen, welche Prozessbäume aus Event Logs generieren [Bui12]. Weitere Techniken wurden in den eigenen Beiträgen [Phi19c] und [Rom19] vorgestellt. Insbesondere im Feld der medizinischen Prozesse hat sich zudem gezeigt, dass sich speziell Prozessbäume gut eignen, um variierende Muster in Behandlungsprozessen aufzudecken [Kir18].

4.2 Prozessbaum als Ausgangsmodell

Ausgangsbasis der Transformation sei ein Prozessbaum $T = (\mathcal{X}, \mathcal{Y})$ nach Definition 3.4 mit Knotenmenge \mathcal{X} und Kantenmenge \mathcal{Y} . Die Knotenmenge \mathcal{X} lasse sich weiter aufteilen in

- \mathcal{A} : die Menge der elementaren Aktionen (Teilmenge der Blattknoten),
- \mathcal{T} : die Menge der stillen Aktionen (Teilmenge der Blattknoten),
- \mathcal{U} : die Menge der Operatorknoten (Menge der inneren Knoten).

Dabei gebe die Hilfsfunktion $f_{\text{Typ}} : \mathcal{U} \rightarrow \{\rightarrow, \times, \wedge, \oslash\}$ für jeden Operatorknoten seinen Typ zurück, also ob es sich bei dem betrachteten Operatorknoten um einen sequentiellen, selektiven, nebenläufigen oder iterativen Verschaltungsoperator handelt. Da für einige Operatorknoten die Reihenfolge der Kinder relevant ist, ist der Prozessbaum T als ein sogenannter gepflanzter Baum [Hoc10] gegeben, bei dem an jedem inneren Knoten $u \in \mathcal{U}$ eine Reihenfolge $\rho(u)$ der direkten Nachfahren von u – also sprich: von dessen Kindern – vorgegeben ist.

Basierend darauf sei eine partielle Hilfsfunktion $f_{\text{Pos}} : (\mathcal{U} \times \mathcal{X}) \rightarrow \mathbb{N}$ gegeben, welche bei Eingabe eines Tupels bestehend aus einem Operatorknoten $u \in \mathcal{U}$

und einem weiteren Knoten $x \in \mathcal{X}$ das Folgende zurückgibt:

$$f_{\text{Pos}}((u, x)) = \begin{cases} r, & \text{falls } (u, x) \in \mathcal{Y} \\ \perp, & \text{sonst} \end{cases} . \quad (4.1)$$

Hierbei gebe $r \in \mathbb{N}$ die Position des betrachteten Kindes x in der Reihenfolge $\rho(u)$ der Kindknoten von u an¹. Dies ist insbesondere für die Operatoren \rightarrow und \mathcal{O} von Interesse. Ist der Knoten x hingegen kein Kindknoten von u , existiert also keine Kante von u nach x , so wird durch die partielle Funktion kein Wert zugewiesen (Symbol \perp).

Weiterhin sei der Prozessbaum als sogenannter knotenorientierter Baum gegeben, bei dem Nutzdaten in jedem Knoten des Baumes abgelegt werden können [Fis18]. Folgende Nutzdaten können so im Baum abgelegt werden:

- $\text{Inp}(a) = \{i_0, \dots, i_l\}$: In jedem Knoten $a \in \mathcal{A}$ kann eine Liste aus $l + 1 \in \mathbb{N}$ Datenobjekten hinterlegt werden, welche während der Ausführung von a als Eingabe benötigt werden. Die Elemente der Liste sind durch die Menge $\text{Inp}(a)$ gegeben.
- $\text{Out}(a) = \{o_0, \dots, o_k\}$: In einem Knoten $a \in \mathcal{A}$ kann eine Liste aus $k + 1 \in \mathbb{N}$ Datenobjekten hinterlegt werden, welche während der Ausführung von a als Ausgabe erzeugt werden. Die Elemente der Liste sind durch die Menge $\text{Out}(a)$ gegeben.

Basierend darauf sei festgelegt:

- Die Menge der Eingaben aller Aktionen: $\mathcal{B}_{\text{Inp}} = \bigcup_{a \in \mathcal{A}} \text{Inp}(a)$,
- Die Menge der Ausgaben aller Aktionen: $\mathcal{B}_{\text{Out}} = \bigcup_{a \in \mathcal{A}} \text{Out}(a)$.

¹ Hinweis: Das Symbol \mathbb{N} bezeichnet die Menge der natürlichen Zahlen einschließlich der 0 (siehe Symbolverzeichnis ab Seite 209).

4.3 FHIR Plan als Zielmodell

Ziel der Transformation sei ein FHIR Plan, welcher durch einen Graphen $H = (\mathcal{K}, \mathcal{M})$ mit Knotenmenge \mathcal{K} und Kantenmenge \mathcal{M} gegeben sei. Die Knotenmenge \mathcal{K} lasse sich partitionieren in die folgenden disjunkten Knotenmengen:

- \mathcal{J} : Menge der elementaren Aktionen,
- \mathcal{J}_τ : Menge der stillen Aktionen,
- \mathcal{G}_\rightarrow : Menge der gruppierenden Aktionen für Sequenzen,
- \mathcal{G}_\times : Menge der gruppierenden Aktionen für Entscheidungen,
- \mathcal{G}_\wedge : Menge der gruppierenden Aktionen für Nebenläufigkeiten,
- $\mathcal{G}_\mathcal{O}$: Menge der gruppierenden Aktionen für Iterationen,
- \mathcal{O}_{Inp} : Menge der Dateninputs,
- \mathcal{O}_{Out} : Menge der Datenoutputs.

Zudem beschreibe $\mathcal{G} = \mathcal{G}_\rightarrow \cup \mathcal{G}_\times \cup \mathcal{G}_\wedge \cup \mathcal{G}_\mathcal{O}$ die Menge der gruppierenden Aktionen. Weiterhin sei die Menge der gerichteten Kanten gegeben durch:

$$\mathcal{M} = \{(e_i, e_j) \mid (e_i, e_j) \in ((\mathcal{K} \setminus (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}})) \times \mathcal{K}, e_i \neq e_j)\}.$$

Es existieren also keine Kanten ausgehend von Dateninputs oder Datenoutputs. Darüber hinaus gibt es keine Kanten von einem Knoten zurück in sich selbst – der Graph ist also frei von Schlingen (schlingenfrei).

Die Kanten \mathcal{M} können weiter unterschieden werden in:

- \mathcal{M}_{Sub} : Menge der Kanten, die gruppierende Aktionen mit ihren Subaktionen verbinden,
- \mathcal{M}_{Vor} : Menge der Kanten, die repräsentieren, dass eine Aktion vor dem Ende einer anderen Aktion ausgeführt werden muss (before-end),
- $\mathcal{M}_{\text{Nach}}$: Menge der Kanten, die repräsentieren, dass eine Aktion nach dem Ende einer anderen Aktion ausgeführt werden muss (after-end),

- \mathcal{M}_{Inp} : Menge der Kanten, die Aktionen mit ihren zugehörigen Dateninputs verbinden,
- \mathcal{M}_{Out} : Menge der Kanten, die Aktionen mit ihren zugehörigen Datenoutputs verbinden.

4.4 Transformation eines Prozessbaums in einen FHIR-Plan

Zur Transformation eines Prozessbaums $T = (\mathcal{X}, \mathcal{Y})$ in einen FHIR-Plan $H = (\mathcal{K}, \mathcal{M})$ werde zunächst eine bijektive Funktion definiert, welche die Knoten des Prozessbaums auf eine Teilmenge der Knoten des FHIR-Plans abbildet:

$$f_{\text{Map}} : \mathcal{X} \rightarrow \mathcal{K} \setminus (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}}). \quad (4.2)$$

Dabei sei die Menge der Dateninputs \mathcal{O}_{Inp} und die Menge der Datenoutputs \mathcal{O}_{Out} nicht in der Zielmenge von f_{Map} enthalten¹.

Aufgrund der Forderung nach Bijektivität gilt nun:

- f_{Map} ist injektiv, d.h. kein Element aus der Zielmenge wird mehrfach getroffen,
- f_{Map} ist surjektiv, d.h. jedes Element aus der Zielmenge wird getroffen,
- es existiert eine Umkehrabbildung f_{Map}^{-1} , d.h. die Funktion f_{Map} ist invertierbar.

Mit der Funktion f_{Map} werden jeder elementaren Aktion $a \in \mathcal{A}$, jeder stillen Aktion $\tau \in \mathcal{T}$ und jedem Operatorknoten $u \in U$ des Prozessbaums eindeutig

¹ Hinweis: Dateninputs und -outputs eines FHIR-Plans werden in einem späteren Teil dieses Abschnittes noch separat beleuchtet.

eine elementare Aktion $j \in \mathcal{J}$, eine stille Aktion $j_\tau \in \mathcal{J}_\tau$ und eine gruppierende Aktion $g \in \mathcal{G}$ des FHIR-Plans zugeordnet:

$$f_{\text{Map}}(x) = \begin{cases} j \in \mathcal{J}, & \text{falls } x \in \mathcal{A} \\ j_\tau \in \mathcal{J}_\tau, & \text{falls } x \in \mathcal{T} \\ g \in \mathcal{G}, & \text{falls } x \in \mathcal{U} \end{cases} . \quad (4.3)$$

Mithilfe von f_{Map} werden die Knoten- und Kantenmengen des FHIR-Plans wie folgt gebildet:

$$\mathcal{J} = \{f_{\text{Map}}(a) \mid a \in \mathcal{A}\}, \quad (4.4)$$

$$\mathcal{J}_\tau = \{f_{\text{Map}}(\tau) \mid \tau \in \mathcal{T}\}, \quad (4.5)$$

$$\mathcal{G}_\rightarrow = \{f_{\text{Map}}(u) \mid u \in \mathcal{U}, f_{\text{Typ}}(u) = \rightarrow\}, \quad (4.6)$$

$$\mathcal{G}_\times = \{f_{\text{Map}}(u) \mid u \in \mathcal{U}, f_{\text{Typ}}(u) = \times\}, \quad (4.7)$$

$$\mathcal{G}_\wedge = \{f_{\text{Map}}(u) \mid u \in \mathcal{U}, f_{\text{Typ}}(u) = \wedge\}, \quad (4.8)$$

$$\mathcal{G}_\emptyset = \{f_{\text{Map}}(u) \mid u \in \mathcal{U}, f_{\text{Typ}}(u) = \emptyset\}, \quad (4.9)$$

$$\mathcal{M}_{\text{Sub}} = \{(f_{\text{Map}}(x_1), f_{\text{Map}}(x_2)) \mid \{x_1, x_2\} \subseteq \mathcal{X}, (x_1, x_2) \in \mathcal{Y}\}, \quad (4.10)$$

$$\mathcal{M}_{\text{Vor}} = \{(f_{\text{Map}}(x_2), f_{\text{Map}}(x_1)) \mid x_2 \in \mathcal{X}, x_1 \in Pa(x_2)\}, \quad (4.11)$$

$$\mathcal{M}_{\text{Nach}} = \{(f_{\text{Map}}(x_1), f_{\text{Map}}(x_2)) \mid \{x_1, x_2\} \subseteq \mathcal{X}, \quad (4.12)$$

$$Pa(x_1) = Pa(x_2) = \{x_{Pa}\}, f_{\text{Typ}}(x_{Pa}) = \rightarrow, \quad (4.13)$$

$$f_{\text{Pos}}(x_2) = f_{\text{Pos}}(x_1) - 1\}. \quad (4.14)$$

Dabei ist die Menge der elementaren Aktionen \mathcal{J} des FHIR-Plans durch das Anwenden der bijektiven Funktion f_{Map} auf die Aktionen \mathcal{A} des Prozessbaums gegeben (4.4). Weiterhin ist die Menge der stillen Aktionen \mathcal{J}_τ im FHIR-Plan durch das Bild der stillen Aktionen \mathcal{T} des Prozessbaums festgelegt (4.5).

In (4.6) - (4.9) werden die Mengen der gruppierenden Aktionen \mathcal{G} des FHIR-Plans durch Anwenden von f_{Map} auf die entsprechenden Operator-knoten des Prozessbaums erzeugt. Dies geschieht jeweils für sequentielle, selektive, nebenläufige sowie iterative Verschaltungen.

Die Kanten \mathcal{M}_{Sub} , welche im FHIR-Plan die Aktionen mit ihren Subaktionen verbinden, werden in (4.10) erstellt. Dafür werden Kanten zwischen jenen Bildern $f_{\text{Map}}(x_1)$ und $f_{\text{Map}}(x_2)$ gezogen, deren Urbilder x_1 und x_2 im Prozessbaum ebenfalls verbunden waren. Sprich: Auch die Bilder adjazenter Knoten bleiben adjazent.

Wie später noch wichtig werden wird, handelt es sich bei der Transformation (4.4)-(4.10) um einen Graphen-Homomorphismus. Gleichzeitig wird aber gezeigt werden, dass sich sämtliche weitere Konstrukte in deterministische Weise aus dem Prozessbaum gewinnen lassen und lediglich die Eigenheiten der Repräsentation des FHIR-Plans widerspiegeln. Zu diesen Eigenheiten gehört beispielsweise, dass im FHIR-Plan neben der Kantenmenge \mathcal{M}_{Sub} zusätzlich weitere Kantenmengen benötigt werden. Diese Mengen und weitere Konstrukte werden in den folgenden Absätzen bis Seite 92 beschrieben.

In (4.11) werden die Kanten \mathcal{M}_{Vor} erzeugt, die repräsentieren, dass eine Aktion vor Beendigung einer anderen ausgeführt werden muss. Im FHIR-Plan werden hierzu Kanten zwischen den Bildern $f_{\text{Map}}(x_2)$ und $f_{\text{Map}}(x_1)$ zweier Knoten x_2 und x_1 des Prozessbaums gezogen. Hierbei kann x_2 ein Aktions- oder Operatorknoten des Prozessbaums sein. Weiterhin muss im Prozessbaum x_1 Elternteil von x_2 gewesen sein, womit implizit folgt, dass x_1 ein Operatorknoten des Prozessbaums ist. Die Kanten in \mathcal{M}_{Vor} verlaufen gerade in entgegengesetzter Richtung der Kanten in \mathcal{M}_{Sub} . Obwohl diese Kanten auf den ersten Blick redundant erscheinen mögen, sind diese allerdings notwendig um eine sinnvolle FHIR-Repräsentation zu erhalten.

Die Kanten $\mathcal{M}_{\text{Nach}}$, die im FHIR-Plan repräsentieren, dass eine Aktion nach dem Ende einer anderen Aktion ausgeführt werden muss, werden in (4.12) - (4.14) erstellt. Dazu werden Kanten zwischen den Bildern $f_{\text{Map}}(x_1)$ und $f_{\text{Map}}(x_2)$ zweier Knoten x_1 und x_2 des Prozessbaums erstellt. Damit eine Kante im FHIR-Plan notwendig wird, müssen die Knoten x_1 und x_2 im Prozessbaum einen gemeinsamen Elternknoten besitzen. Dieser muss vom Typ eines sequentiellen Verschaltungsoperators sein (4.13) – repräsentiert wird dies durch das Symbol \rightarrow . Zudem spielt die Reihenfolge der Kinder des sequentiellen Verschaltungsoperators eine Rolle, was in (4.14) berücksichtigt wird: Sind x_1 und x_2 im Prozessbaum benachbarte Kinder und liegt x_1 rechts

von x_2 , so wird im FHIR Plan eine Kante zwischen den Bildern $f_{\text{Map}}(x_1)$ und $f_{\text{Map}}(x_2)$ erstellt.

Zur Vervollständigung der Transformation sei weiterhin eine zweite bijektive Funktion

$$f_{\text{Map, Dat}} : (\mathcal{B}_{\text{Inp}} \cup \mathcal{B}_{\text{Out}}) \rightarrow (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}}) \quad (4.15)$$

gegeben, welche die Menge aller Ein- und Ausgabe-Datenobjekte des Prozessbaums eindeutig auf die Menge aller Dateninputs und Datenoutputs des FHIR-Plans abbildet:

$$f_{\text{Map, Dat}}(x) = \begin{cases} o_{\text{Inp}} \in \mathcal{O}_{\text{Inp}}, & \text{falls } x \in \mathcal{B}_{\text{Inp}} \\ o_{\text{Out}} \in \mathcal{O}_{\text{Out}}, & \text{falls } x \in \mathcal{B}_{\text{Out}} \end{cases} . \quad (4.16)$$

Mithilfe von $f_{\text{Map, Dat}}(x)$ lassen sich nun noch die folgenden Knoten- und Kantenmengen des FHIR-Plans erzeugen:

$$\mathcal{O}_{\text{Inp}} = \{ f_{\text{Map, Dat}}(b) \mid b \in \mathcal{B}_{\text{Inp}} \}, \quad (4.17)$$

$$\mathcal{O}_{\text{Out}} = \{ f_{\text{Map, Dat}}(b) \mid b \in \mathcal{B}_{\text{Out}} \}, \quad (4.18)$$

$$\mathcal{M}_{\text{Inp}} = \{ (f_{\text{Map, Dat}}(a), f_{\text{Map, Dat}}(b)) \mid a \in \mathcal{A}, b \in \text{Inp}(a) \}, \quad (4.19)$$

$$\mathcal{M}_{\text{Out}} = \{ (f_{\text{Map, Dat}}(a), f_{\text{Map, Dat}}(b)) \mid a \in \mathcal{A}, b \in \text{Out}(a) \}. \quad (4.20)$$

Hierbei wird in (4.17) und (4.18) für jedes Datenelement aus dem Prozessbaum ein entsprechender Dateninput oder -output im FHIR-Plan erstellt. Da die Funktion $f_{\text{Map, Dat}}$ bijektiv ist, gibt es jeweils eine eindeutige Zuordnung zwischen den Mengen \mathcal{B}_{Inp} und \mathcal{B}_{Out} des Prozessbaums und den Mengen \mathcal{O}_{Inp} und \mathcal{O}_{Out} des FHIR-Plans.

In (4.19) und (4.20) werden anschließend noch die Kanten erstellt, welche die Dateninputs und die Datenoutputs mit den zugehörigen Aktionen $j \in \mathcal{J}$ im FHIR-Plan verbinden. Dabei wird zwischen einer Aktion $j = f_{\text{Map, Dat}}(a)$ sowie einem Dateninput oder aber Datenoutput $f_{\text{Map, Dat}}(b)$ dann eine Kante gezogen, wenn das Urbild b im Prozessbaum ein der Aktion $a \in \mathcal{A}$ zugeordnetes Datenelement der Eingabe oder aber der Ausgabe war.

4.5 Transformation eines FHIR-Plans in einen Prozessbaum

Zur Transformation eines FHIR-Plans $H = (\mathcal{K}, \mathcal{M})$ in einen Prozessbaum $T = (\mathcal{X}, \mathcal{Y})$ wird auf die Umkehrfunktion der in (4.2) und (4.3) definierten bijektiven Funktion f_{Map} zurückgegriffen:

$$f_{\text{Map}}^{-1} : (\mathcal{K} \setminus (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}})) \rightarrow \mathcal{X}. \quad (4.21)$$

Die Funktion bildet eine Teilmenge der Knoten des FHIR-Plans – alle Knoten außer den Dateninputs und Datenoutputs – auf die Knoten des Prozessbaums ab:

$$f_{\text{Map}}^{-1}(k) = \begin{cases} a \in \mathcal{A}, & \text{falls } k \in \mathcal{J} \\ \tau \in \mathcal{T}, & \text{falls } k \in \mathcal{J}_\tau \\ u \in \mathcal{U}, & \text{falls } k \in \mathcal{G} \end{cases}. \quad (4.22)$$

Mithilfe von f_{Map}^{-1} werden die Knoten- und Kantenmengen des Prozessbaums wie folgt gebildet:

$$\mathcal{A} = \{f_{\text{Map}}^{-1}(j) \mid j \in \mathcal{J}\}, \quad (4.23)$$

$$\mathcal{T} = \{f_{\text{Map}}^{-1}(j_\tau) \mid j_\tau \in \mathcal{J}_\tau\}, \quad (4.24)$$

$$\mathcal{U} = \{f_{\text{Map}}^{-1}(g) \mid g \in \mathcal{G}\}, \quad (4.25)$$

$$\mathcal{Y} = \{(f_{\text{Map}}^{-1}(x_1), f_{\text{Map}}^{-1}(x_2)) \mid (x_1, x_2) \in \mathcal{M}_{\text{Sub}}\}. \quad (4.26)$$

Alle elementaren Aktionen $j \in \mathcal{J}$ des FHIR-Plans werden dabei in die Aktionen $a \in \mathcal{A}$ des Prozessbaums übersetzt (4.23). Die stillen Aktionen $j_\tau \in \mathcal{J}_\tau$ des FHIR-Plans werden analog auf die stillen Aktionen $\tau \in \mathcal{T}$ des Prozessbaums abgebildet (4.24). Auch zwischen der Menge der gruppierenden Aktionen \mathcal{G} des FHIR-Plans und der Menge der Operatorknoten \mathcal{U} des Prozessbaums besteht eine Eins-zu-Eins-Beziehung (4.25).

In (4.26) wird die Kantenmenge \mathcal{Y} des Prozessbaums erzeugt: Für jede Kante, die im FHIR-Plan eine Aktion mit ihren Subaktionen verbindet wird auch im

Prozessbaum eine Kante zwischen den entsprechenden Bildern dieser Aktion und ihrer Subaktionen hinzugefügt.

Weiterhin werde für die Operatorknoten des Prozessbaums der entsprechende Typ gesetzt:

$$f_{\text{Typ}}(f_{\text{Map}}^{-1}(g)) = \begin{cases} \rightarrow, & \text{falls } g \in \mathcal{G}_{\rightarrow} \\ \wedge, & \text{falls } g \in \mathcal{G}_{\wedge} \\ \times, & \text{falls } g \in \mathcal{G}_{\times} \\ \emptyset, & \text{falls } g \in \mathcal{G}_{\emptyset} \end{cases}. \quad (4.27)$$

Zum Schluss werden noch die in den Knoten des Baums gespeicherten Datenelemente erzeugt. Hierzu wird auf die Umkehrabbildung der in (4.15) und (4.16) definierten Funktion $f_{\text{Map, Dat}}$ zurückgegriffen:

$$f_{\text{Map, Dat}}^{-1} : (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}}) \rightarrow (\mathcal{B}_{\text{Inp}} \cup \mathcal{B}_{\text{Out}}). \quad (4.28)$$

Die Funktion bildet die Mengen aller Dateninputs und Datenoutputs des FHIR-Plans eindeutig auf die Mengen aller Datenobjekte der Eingabe und Ausgabe des Prozessbaums ab:

$$f_{\text{Map, Dat}}^{-1}(k) = \begin{cases} b_{\text{Inp}} \in \mathcal{B}_{\text{Inp}}, & \text{falls } k \in \mathcal{O}_{\text{Inp}} \\ b_{\text{Out}} \in \mathcal{B}_{\text{Out}}, & \text{falls } k \in \mathcal{O}_{\text{Out}} \end{cases}. \quad (4.29)$$

Mithilfe von $f_{\text{Map, Dat}}^{-1}$ können nun die Datenobjekte des Prozessbaums erzeugt und für jede Aktion a in die Mengen der Eingabelemente $\text{Inp}(a)$ und der Ausgabelemente $\text{Out}(a)$ zusammengefasst werden:

$$\forall a \in \mathcal{A}, a = f_{\text{Map, Dat}}^{-1}(j) : \quad (4.30)$$

$$\text{Inp}(a) = \{ f_{\text{Map, Dat}}^{-1}(o) \mid (o, j) \in \mathcal{M}, o \in \mathcal{O}_{\text{Inp}} \},$$

$$\text{Out}(a) = \{ f_{\text{Map, Dat}}^{-1}(o) \mid (j, o) \in \mathcal{M}, o \in \mathcal{O}_{\text{Out}} \}. \quad (4.31)$$

4.6 Erhaltung der Struktur der Originalmodelle

Die Transformation eines Modells in eine alternative Repräsentation sollte dessen Struktur so gut wie möglich erhalten, damit anschließend eine verlustfreie Rücktransformation möglich ist.

In diesem Kapitel wird daher untersucht, inwiefern die beschriebenen Transformationen die Struktur der Ursprungsmodelle erhalten. Da Ausgangs- und Zielmodelle jeweils als Graphen dargestellt werden können, wird in einem ersten Schritt überprüft, welche Eigenschaften die Abbildung des einen Graphen in den anderen Graphen aufweist. In diesem Kontext werden Graph-Homomorphismen und -Isomorphismen als spezielle, strukturerhaltende Abbildungen zwischen Graphen definiert.

4.6.1 Strukturerhaltung bei Hintransformation

Bei der Hintransformation wird ausgehend von einem Prozessbaum $T = (\mathcal{X}, \mathcal{Y})$ ein FHIR-Plan $H = (\mathcal{K}, \mathcal{M})$ erzeugt. Im Rahmen dieser Transformation wird zunächst die Knotenmenge \mathcal{X} des Prozessbaums mithilfe der in 4.2 definierten bijektiven Funktion f_{Map} in die Knotenmenge des FHIR-Plans abgebildet:

$$f_{\text{Map}} : \mathcal{X} \rightarrow \mathcal{K} \setminus (\mathcal{O}_{\text{inp}} \cup \mathcal{O}_{\text{out}}). \quad (4.32)$$

Dabei umfasst der Bildbereich von f_{Map} nur eine Teilmenge der Knotenmenge des FHIR-Plans – alle Knoten außer den Dateninputs und den Datenoutputs. Die weitere Betrachtung fokussiert sich daher auf den Kontrollfluss.

Durch f_{Map} wird gemäß (4.3) jeder Operatorknoten des Prozessbaums eindeutig auf eine gruppierende Aktion des FHIR-Plans abgebildet. Gleichzeitig wird ebenso jede Aktion des Prozessbaums – sei es eine elementare oder stille Aktion – eindeutig auf eine entsprechende Aktion des FHIR-Plans abgebildet.

Damit diese Transformation strukturerhaltend ist, müssen die Eltern-Kind-Beziehungen des Prozessbaums T auch im Graphen des FHIR-Plans H zu finden sein. Elternknoten im Prozessbaum sind gemäß Definition immer Operatorknoten (vgl. Kapitel 3.4). Betrachtet man einen beliebigen Operatorknoten $u \in \mathcal{U}$ des Prozessbaums, so sind ihm diverse Kindknoten $x \in \mathcal{X}$ zugeordnet, die jeweils über eine Kante aus der Kantenmenge \mathcal{Y} des Prozessbaums mit dem Operatorknoten u verbunden sind:

$$Chi(u) = \{x \in \mathcal{X} \mid \exists(u, x) \in \mathcal{Y}\}. \quad (4.33)$$

Zur Erhaltung der Eltern-Kind-Beziehung muss für einen beliebigen Operatorknoten $u \in \mathcal{U}$ des Prozessbaums, sowie dessen Kinder x gelten:

$$\forall x \in \mathcal{X}, \forall u \in \mathcal{U} : x \in Chi(u) \Rightarrow f_{\text{Map}}(x) \in Chi(f_{\text{Map}}(u)). \quad (4.34)$$

Die transformierten Kindknoten von u müssen also im FHIR-Plan wiederum Kindknoten des Bildes $f_{\text{Map}}(u)$ sein. Dabei kann die Menge der transformierten Kindknoten $\{f_{\text{Map}}(x) \mid x \in \mathcal{X}\}$ auch nur eine Teilmenge der Kindknoten von $f_{\text{Map}}(u)$ ausmachen:

$$\{f_{\text{Map}}(x) \mid x \in \mathcal{X}\} \subseteq Chi(f_{\text{Map}}(u)). \quad (4.35)$$

Zur Prüfung des Sachverhalts (4.34) sind insbesondere die Kanten \mathcal{M}_{Sub} des FHIR-Plans von Interesse, da diese im FHIR-Plan Eltern-Kind-Beziehungen zwischen Aktionen und ihren Subaktionen repräsentieren. Die Kantenmenge \mathcal{M}_{Sub} wird unter Anwendung der Funktion f_{Map} erzeugt, indem zwischen den Bildern zweier adjazenter Knoten eine Kante gezogen wird:

$$\mathcal{M}_{\text{Sub}} = \{(f_{\text{Map}}(x_1), f_{\text{Map}}(x_2)) \mid x_1, x_2 \in \mathcal{X}, (x_1, x_2) \in \mathcal{Y}\}. \quad (4.36)$$

Dass Aussage (4.34) für die in (4.4) - (4.14) definierte Transformation gilt, wird im Folgenden bewiesen:

- Gegeben seien zwei Knoten $x \in \mathcal{X}$ und $u \in \mathcal{U}$ des Prozessbaums mit $x \in Chi(u)$. Nach Voraussetzung existieren dann Kanten $(u, x) \in \mathcal{Y}$, vgl. (4.33).

- Weiterhin gilt, dass der Definitionsbereich der Funktion f_{Map} ganz \mathcal{U} umfasst, vgl. (4.32), und f_{Map} bijektiv ist. Demnach existiert zu jedem $u \in \mathcal{U}$ ein eindeutiges Bild $f_{\text{Map}}(u) \in \mathcal{K}$ in der Knotenmenge des FHIR-Plans.
- Da f_{Map} die Operatorknoten des Prozessbaums auf die Menge der gruppierenden Aktionen $\mathcal{G} \subseteq \mathcal{K}$ abbildet, vgl. (4.3), existiert genauer zu jedem $u \in \mathcal{U}$ auch ein eindeutiges $g = f_{\text{Map}}(u) \in \mathcal{G}$.
- Da f_{Map} zudem die Knoten für Aktionen und stille Aktionen des Prozessbaums eindeutig auf die Menge der elementaren und stillen Aktionen \mathcal{J} und \mathcal{J}_τ des FHIR-Plans abbildet, vgl. (4.3), existiert allgemein zu jedem Kind x von u , d.h. zu jedem $x \in \text{Chi}(u)$, auch ein eindeutiges Bild $k = f_{\text{Map}}(x) \in \mathcal{K}$.
- Nun ist noch zu zeigen, dass die Bilder der Kindknoten $k = f_{\text{Map}}(x)$ auch im FHIR-Plan durch Kanten mit den Bildern der Elternknoten $g = f_{\text{Map}}(u)$ verbunden sind.
- Bei der Transformation wird die Kantenmenge \mathcal{M}_{Sub} des FHIR-Plans gemäß (4.36) gebildet. Demnach sind zwei Knoten des FHIR-Plans durch eine Kante verbunden, sofern auch deren Urbilder im Prozessbaum adjazent sind. Da $(u, x) \in \mathcal{Y}$, d. h. u und x im Prozessbaum adjazent sind, existiert auch im FHIR-Plan eine Kante $(g, k) = (f_{\text{Map}}(u), f_{\text{Map}}(x)) \in \mathcal{M}_{\text{Sub}}$ für alle Paare aus Elternknoten u und Kindknoten x des Prozessbaums.
- Existiert nun eine Kante von einem Knoten g zu einem Knoten k des FHIR-Plans, so ist dies genau die Definition einer Eltern-Kind-Beziehung. Es gilt also: $k = f_{\text{Map}}(x) \in \text{Chi}(g) = \text{Chi}(f_{\text{Map}}(u))$.

Letztlich lässt sich feststellen, dass es sich bei der in (4.4) - (4.14) gegebenen Transformation eines Prozessbaums in einen FHIR-Plan um einen sogenannten Graphen-Homomorphismus handelt. Graphen-Homomorphismen

sind spezielle, strukturerhaltende Abbildungen zwischen Graphen¹. Existiert jeweils eine solche strukturerhaltende Abbildungen für die Hin- und die Rückrichtung der Transformation zwischen zwei Graphen, dann nennen sich die beiden Graphen homomorph-äquivalent.

Ein Graphen-Homomorphismus kann dabei wie folgt definiert werden [Kna15]:

Gegeben seien zwei Graphen $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ und $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$ mit den Knotenmengen \mathcal{V}_1 und \mathcal{V}_2 , sowie den Kantenmengen \mathcal{E}_1 und \mathcal{E}_2 . Eine Abbildung $f_{\text{Hom}} : \mathcal{V}_1 \rightarrow \mathcal{V}_2$ (auch $f_{\text{Hom}} : G_1 \rightarrow G_2$ geschrieben²) heißt Graphen-Homomorphismus³, wenn sie kantenerhaltend ist, d.h. wenn gilt

$$\forall x, y \in \mathcal{V}_1 : (x, y) \in \mathcal{E}_1 \Rightarrow (f_{\text{Hom}}(x), f_{\text{Hom}}(y)) \in \mathcal{E}_2. \quad (4.37)$$

Zwischen den Bildern zweier im Ursprungsgraphen adjazenter Knoten wird im Zielgraph also auch immer eine Kante gezogen. Durch diese Forderung induziert ein Graphen-Homomorphismus f_{Hom} letztlich auch immer eine Abbildung zwischen den Kantenmengen des Ursprungs- und des Zielgraphen, $f_{\text{Hom, Kanten}} : \mathcal{E}_1 \rightarrow \mathcal{E}_2$, sodass – vereinfacht gesagt – das Folgende gilt: Durch einen Homomorphismus werden Kanten auf Kanten abgebildet.

Da die Kantenmenge des Prozessbaums lediglich durch Eltern-Kind-Beziehungen gegeben ist und diese wie oben gezeigt vollständig in die Kantenmenge \mathcal{M}_{Sub} des Prozessbaums abgebildet werden, ist die Transformation von dem Prozessbaum in den FHIR-Plan also ein strukturerhaltender Homomorphismus.

¹ Hinweis: Strukturerhaltende Abbildungen zwischen mathematischen Strukturen werden allgemein als Homomorphismen bezeichnet. Die Homomorphismen der Vektorräume sind beispielsweise die linearen Abbildungen.

² Hinweis: Da die Abbildung zwischen den Knotenmengen der beiden Graphen gemeinsam mit der Forderung nach Kantenerhaltung letztlich auch eine Abbildung zwischen den beiden Kantenmengen induziert, spricht man bei einem Graphen-Homomorphismus f_{Hom} auch von einer Abbildung zwischen zwei Graphen. Man kann den Homomorphismus dann auch als Abbildung zwischen den disjunkten Vereinigungen der Knoten- und Kantenmengen der Graphen definieren: $f_{\text{Hom}} : (\mathcal{V}_1 + \mathcal{E}_1) \rightarrow (\mathcal{V}_2 + \mathcal{E}_2)$, vgl. z.B. [Tin76].

³ Hinweis: Wird teils auch als Graphen-Morphismus bezeichnet.

Betrachtet man einen Teilgraphen¹ des FHIR-Plans, kann sogar gezeigt werden, dass der Teilgraph und der Prozessbaum isomorph sind, d.h. ein sogenannter Graphen-Isomorphismus existiert. Ein Graphen-Isomorphismus unterscheidet sich von einem Graphen-Homomorphismus dadurch, dass zusätzlich Bijektivität gefordert wird, sprich: Auch die Abbildung in umgekehrter Richtung – also vom Zielgraphen wieder zurück in den Ursprungsgraphen – muss ein Graphen-Homomorphismus sein.

Dadurch folgt für die vom Isomorphismus induzierte Abbildung $f_{\text{Hom, Kanten}}$ zwischen den Kantenmengen der beiden Graphen die Surjektivität: Ist eine Kante im Zielgraphen vorhanden, so muss es im Ursprungsgraphen ein entsprechendes Urbild dieser Kante gegeben haben.

Dementsprechend gilt bei einem Graphen-Isomorphismus $f_{\text{Iso}} : \mathcal{V}_1 \rightarrow \mathcal{V}_2$:

$$\forall x, y \in \mathcal{V}_1 : (x, y) \in \mathcal{E}_1 \Leftrightarrow (f_{\text{Iso}}(x), f_{\text{Iso}}(y)) \in \mathcal{E}_2. \quad (4.38)$$

Betrachtet man einen Teilgraphen $\tilde{H} = (\tilde{\mathcal{K}}, \tilde{\mathcal{M}})$ des FHIR-Plans mit der Knotenmenge $\tilde{\mathcal{K}} = (\mathcal{K} \setminus (\mathcal{O}_{\text{Imp}} \cup \mathcal{O}_{\text{Out}}))$ und der Kantenmenge M_{Sub} , sind T und \tilde{H} isomorph und f_{Map} ist ein Graphenisomorphismus. Hierauf wird im nächsten Kapitel 4.6.2, bei der Untersuchung der Strukturhaltung bei Rücktransformation, aufgebaut.

Weiter ist nun zu zeigen, dass sich auch die noch fehlenden Knoten und Kanten des FHIR-Plans eindeutig aus dem Prozessbaum und dem schon erzeugten Teilgraphen \tilde{H} ableiten lassen.

Zunächst werden die Kanten M_{Vor} betrachtet. Diese Kanten repräsentieren, dass eine Aktion vor dem Ende einer anderen Aktion ausgeführt werden muss. Da diese Kanten im Falle des FHIR-Plans immer in umgekehrter Richtung der Kanten M_{Sub} verlaufen, können sie leicht erstellt werden, indem der Teilgraph \tilde{H} transponiert wird. Hierdurch wird der Graph \tilde{H}^T mit gleicher Knotenmenge $\tilde{\mathcal{K}}$, aber angepasster Kantenmenge erzeugt. Letztere wird gebildet, indem

¹ Hinweis: Ein Teilgraph entsteht aus einem Graphen, indem Knoten und/oder Kanten entfernt werden.

alle Kanten der Menge M_{Sub} umgekehrt werden, wodurch die neue Kantenmenge gerade der Menge M_{Vor} entspricht. Anschließend kann dann die Vereinigung der beiden Teilgraphen $\tilde{H} \cup \tilde{H}^T$ gebildet werden. Resultat ist ein Graph, welcher aus der Knotenmenge $\tilde{\mathcal{K}}$, sowie der Kantenmenge $(M_{\text{Sub}} \cup M_{\text{Vor}})$ besteht.

Das Bilden der Kantenmenge M_{Nach} gemäß (4.12) – (4.14) ist anspruchsvoller. Hierbei muss zum Einen der Typ des jeweiligen Operatorknotens vorliegen, da dererlei Kanten nur zwischen Kindknoten des sequentiellen Verschaltungsoperators erzeugt werden müssen. Zum Anderen muss auch die Reihenfolge der Kindknoten bekannt sein, da diese bei der Verschaltung von Relevanz ist und bestimmt, welche Kindknoten im FHIR-Plan adjazent sind.

Letztlich sind diese beiden Informationen aber eindeutig durch den Prozessbaum vorgegeben. Jedem Elternknoten kann gemäß der Funktion $f_{\text{Typ}} : \mathcal{U} \rightarrow \{\rightarrow, \times, \wedge, \oslash\}$ eindeutig einer der vier Operatortypen zugeordnet werden. Weiterhin ist auch eine eindeutige Reihenfolge der Kindknoten gegeben und bekannt, da es sich bei einem Prozessbaum wie in Kapitel 4.2 beschrieben um einen gepflanzten Baum handelt.

4.6.2 Strukturhaltung bei Rücktransformation

Bei der Rücktransformation wird ausgehend von einem FHIR-Plan $H = (\mathcal{K}, \mathcal{M})$ ein Prozessbaum $T = (\mathcal{X}, \mathcal{Y})$ erzeugt. Hierfür wird zunächst eine Teilmenge der Knoten des FHIR-Plans in die Knotenmenge des Prozessbaums abgebildet. Dies geschieht mithilfe der in 4.21 Umkehrfunktion zu f_{Map} :

$$f_{\text{Map}}^{-1} : (\mathcal{K} \setminus (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}})) \rightarrow \mathcal{X}. \quad (4.39)$$

Dabei umfasst der Definitionsbereich von f_{Map} alle Knoten des FHIR-Plans, außer den Dateninputs und den Datenoutputs, welche später gesondert betrachtet werden. Durch diese Einschränkung auf eine Teilknotenmenge des FHIR-Plans, ist die Abbildung f_{Map} bijektiv und die Umkehrabbildung f_{Map}^{-1} kann gebildet werden. Demnach ist die Funktion f_{Map}^{-1} auch surjektiv, d.h. jedes Element aus der Knotenmenge \mathcal{X} des Prozessbaums wird getroffen.

Wie im vorherigen Kapitel gezeigt, sind sowohl die Abbildung f_{Map} als auch f_{Map}^{-1} Graphen-Homomorphismen bezüglich des Graphens des Prozessbaums T und des Teilgraphens \tilde{H} des FHIR-Plans. \tilde{H} besteht dabei aus der Knotenmenge $\tilde{\mathcal{K}} = (\mathcal{K} \setminus (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}}))$ und der Kantenmenge M_{Sub} . Somit existiert ein Graphen-Isomorphismus und damit jeweils strukturerhaltende Hin- und Rücktransformationen zwischen den beiden Graphen.

Um die Datenelemente zu erzeugen, die in den Knoten des Prozessbaums gespeichert werden, wird die in (4.29) definierte Funktion $f_{\text{Map, Dat}}^{-1}$ genutzt. Da diese wiederum bijektiv ist, ist eine eindeutige Rücktransformation aus den Knoten des FHIR-Plans möglich.

5 **Assistenz bei chirurgischen Interventionen**

In der modernen Medizin wird der Einsatz computergestützter Assistenz immer wichtiger, vgl. [Phi16b, Lal14]. So wird der weltweite Markt für medizinische Robotik und computergestützte chirurgische Ausrüstung bis 2025 auf umgerechnet 9,6 Milliarden Euro anwachsen (bei einer fünfjährigen durchschnittlichen jährlichen Wachstumsrate von 13,5%) [McW20].

Im Rahmen der Computerassistierten Chirurgie (engl. Computer Assisted Surgery) (CAS) können Assistenzfunktionen umgesetzt werden, die Entscheidungsunterstützungen für Mediziner [Ken15] bereitstellen. Eine solche Entscheidungsunterstützung eröffnet Optimierungspotentiale hinsichtlich der Vermeidung falscher Behandlungen beispielsweise durch Verhinderung oder Vermeidung von Flüchtigkeitsfehlern sowie Irrtümern.

In diesem Zusammenhang spielt die chirurgische Phasenerkennung eine wichtige Rolle: Erst durch die Beurteilung des Verlaufs einer laufenden Operation ist eine maßgeschneiderte (d. h. kontextsensitive) und interaktive Entscheidungsunterstützung während eines chirurgischen Eingriffs möglich. Dabei findet nicht nur eine passive Verbreitung (z. B. Verbreitung über Printmedien) von Hilfsmitteln (z. B. medizinische Leitlinien) statt, die nur geringe Auswirkungen auf das tatsächliche Behandlungsverhalten hat [Fie92, Ste11]. Darüber hinaus erlaubt eine maßgeschneiderte Entscheidungshilfe die Filterung tatsächlich verfügbarer Informationen, um operativ effektiv zu sein, indem sie eine Informationsüberflutung von Ärztinnen und Ärzten verhindert [Joy83, Kra13, Lin13].

Die Optimierung eines Arbeitsablaufs, z. B. durch den Vergleich aufgezeichneter Daten mit einem vordefinierten Zielmodell, ist ein weiteres Anwendungsbeispiel für eine Phasenerkennung. Dies ist besonders relevant, da Operationen als eine wichtige Quelle für die Verbesserung der Effizienz des Krankenhauses [Her03] identifiziert werden konnten.

Chirurgische Interventionen lassen sich neben der Dringlichkeit, den betroffene Körperstellen, dem Zweck auch nach der sogenannten Invasivität einteilen. Während bei der minimal-invasiven Chirurgie nur über kleine, künstlich geschaffene Öffnungen in den Körper vorgedrungen wird, sind bei der offenen Chirurgie weitreichendere (invasivere) Methoden notwendig, um das Operationsgebiet, den sogenannten Situs, geeignet zu eröffnen und für eine Behandlung darzustellen [Psc13].

Im folgenden Kapitel 5.1 wird je ein prototypischer Vertreter des minimal-invasiven und des offenen Ansatzes erläutert. In Kapitel 5.2 werden die den beiden Ansätzen zugrunde liegenden Konstituenten des Problems betrachtet und darauf aufbauend in Kapitel 5.3 bereits bekannte Lösungsansätze beleuchtet. Die eigenen Beiträge zur Erweiterung des Standes der Technik werden in den daran anschließenden Kapiteln ausführlich dargestellt.

5.1 Medizinischer Hintergrund

In den folgenden beiden Kapiteln 5.1.1 und 5.1.2 werden die medizinischen Grundlagen zum Verständnis der Versorgung von Patienten mit einem künstlichen Hüftgelenk sowie der chirurgischen Entfernung der Gallenblase erläutert. Diese Eingriffe sind prototypische Vertreter für eine Vielzahl anderer Eingriffe im medizinischen Bereich.

5.1.1 Totale Endoprothese der Hüfte

Die endoprothetische Versorgung eines Patienten mittels einer totalen Endoprothese (TEP) der Hüfte, auch kurz Hüft-TEP¹ genannt, ist ein Eingriff, bei dem das Hüftgelenk des Menschen vollständig (= total) durch eine Prothese, also ein künstliches Hüftgelenk, ersetzt wird². Diese Prothese wird in den Menschen eingebracht (= endo, innen) und verbleibt dort also sogenannte Endoprothese. Gründe für eine solche endoprothetische Versorgung sind beispielsweise eine fortgeschrittene Hüftgelenksarthrose oder der Bruch des Oberschenkelhalses [Fre06, Lür17].

Mit über die Jahren steigenden Fallzahlen³ ist sie aktuell mit etwa 240.000 Eingriffen pro Jahr in Deutschland eine sehr häufige Versorgungsmaßnahme in Krankenhäusern. Auch international ist Deutschland verglichen mit anderen OECD Ländern, Spitzenreiter bei der Implantation künstlicher Hüftgelenke [OEC19]. Werden solcherlei Eingriffe geeignet technisch unterstützt (vgl. Kapitel 5), so ist zu erwarten, dass dies einen Einfluss auf eine Vielzahl an Patienten und ärztlichem Personal haben wird.

Auch wenn es mehrere, gleichermaßen erprobte, Zugangswege im Rahmen des Eingriffs gibt [Nog17, Lür17], kann der Zugang nach Bauer [Bau79] durch den Glutealmuskel (Gesäßmuskel) als eine der etablierten Vorgehensweisen in Deutschland angesehen werden. Für das standardisierte Vorgehen kann ein Soll-Operationsverlauf aufgestellt werden, der teils in entsprechenden Handbüchern beschrieben und mit zusätzlichem Erfahrungswissen der Experten, auch über einzelne Fallbeispiele, angereichert wird, vgl. [Lür17].

Dieser Soll-Operationsverlauf ist aufgrund der standardisierten Vorgehensweise zwischen Personen und Krankenhäusern zunächst einmal übertragbar. Ähnlich wie bei der Thematik der medizinischen Leitlinien [Phi15b, Phi15c,

¹ Hinweis: Der Begriff TEP findet auch bei anderen Körperstrukturen Anwendung. Beispielsweise wird bei Ersetzung des Kniegelenkes von einer Knie-TEP gesprochen.

² Hinweis: Der Mensch verfügt über ein rechtes und ein linkes Hüftgelenk. Egal ob das rechte oder aber das linke Hüftgelenk ersetzt wird, wird gleichermaßen von einer Hüft-TEP gesprochen.

³ Hinweis: Im Jahr 2005 bzw. 2018 wurden 194.453 bzw. 239.204 Eingriffe vorgenommen [Sta19a].

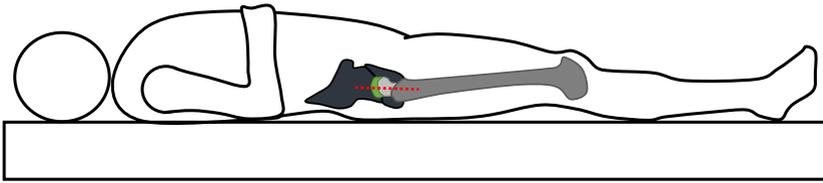


Abbildung 5.1: Lagerung eines Patienten auf dem Rücken zur Durchführung einer endoprothetischen Versorgung mittels Hüft-TEP mit Zugang nach Bauer [Bau79]. Der Zugang erfolgt (siehe rote Markierung) durch schichtweises Freilegen des Hüftgelenks direkt lateral (seitlich) und transgluteal (durch den Glutealmuskel). In Anlehnung an [Lür17].

Phi16a, Phi16b] sollen die Vorgehensweisen aber durch Experten entsprechend den Rahmenbedingungen (beispielsweise Ausstattung des betreffenden Krankenhauses) angepasst und fallspezifisch interpretiert werden.

Einen stark vereinfachten Überblick der endoprothetischen Versorgung mit einer Hüft-TEP nach Bauer [Bau79] zeigen die Abbildungen 5.1, 5.2 und 5.3. Zunächst wird am rechten Bein ein Hautschnitt lateral¹, also seitlich des Beines, durchgeführt (siehe rote Markierung in Abbildung 5.1)².

Der Zugangsweg zum rechten Hüftgelenk erfolgt dann durch schichtweises Freipräparieren. Dies geschieht direkt, also durch den dort liegenden Glutealmuskel hindurch (transgluteal). Die Vorgehensweise wird deswegen teils bezeichnet als (vgl. [Nog17]): direkter lateraler, transglutealer Zugang nach Bauer.

Wie in Abbildung 5.1 und 5.2 dargestellt, besteht der Bereich von Interesse aus:

¹ lat. latus = Flanke, Seite

² Hinweis: In dieser Ausarbeitung werden jene anatomische Lagebezeichnungen verwendet, die invariant bezüglich der Positionierung des Patienten sind. Beispielsweise kranial (schädelwärts) statt superior (oben). Letzteres unterscheidet sich bei einem stehenden und einem liegenden Patienten.

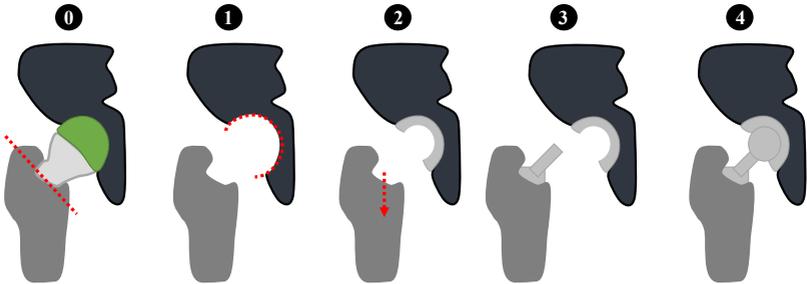


Abbildung 5.2: Stark vereinfachte, skizzenhafte Darstellung der endoprothetischen Versorgung mit einer Hüft-TEP in Anlehnung an [Fre06]. Zunächst wird der Schenkelhals (hellgrau) mit einer Säge durchtrennt um den Hüftkopf (grün) entfernen zu können (0). Danach wird die Hüftpfanne mit einer kugelförmigen, rotierenden, elektrischen Fräse bearbeitet (1). In die Fräsung wird die künstliche Pfanne hineingeschlagen (2). Schließlich wird der Markraum des Oberschenkelknochens freipräpariert (3), der künstliche Schaft eingebracht (4), der künstliche Hüftkopf aufgesetzt sowie schließlich Kopf und Pfanne reponiert, also das nun künstliche Gelenk eingerenkt (5).

- dem Oberschenkelknochen (dunkelgraue Struktur),
- dem Schenkelhals (hellgrauer Bereich),
- dem Hüftkopf (grüner Bereich),
- dem Hüftpfanne bzw. Becken (schwarzer Bereich).

Um den Hüftkopf entfernen zu können, muss zunächst die Gelenkkapsel freipräpariert werden. Danach wird bei der sogenannten Schenkelhalsosteotomie¹ der Schenkelhals mit einer Säge durchtrennt, vgl. (0) in Abbildung 5.2. Der Hüftkopf kann dann mit einem Korkenzieherinstrument geborgen werden. Damit dies gelingen kann, muss nach der Präparation der Gelenkkapsel aber vor der eigentlichen Schenkelhalsosteotomie das Hüftgelenkt luxiert, d. h. ausgekugelt werden.

¹ Hinweis: Osteotomie bezeichnet allgemein das Durchtrennen von Knochen oder Entfernen von Knochenfragmenten.

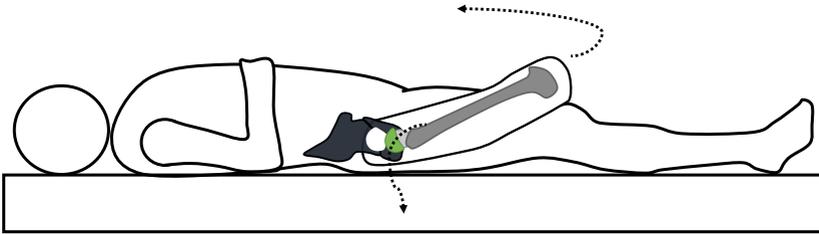


Abbildung 5.3: Schematische Darstellung des ausgelenkten (luxierten) Hüftgelenks. Die Beine sind rechtwinklig zueinander, in der sogenannten Viererposition angeordnet. Der Schenkelhals ist mit einer hellgrauen Fläche markiert. Der Hals wird im Rahmen der Schenkelhalsosteotomie mit einer Säge durchtrennt um den darauf befindlichen Hüftkopf (grün) entfernen zu können.

Nun wird die Hüftpfanne mit einer kugelförmigen Fräse ausgefräst, sowie danach die künstliche Hüftpfanne mit einem Einschlaginstrument und Hammer in den ausgefrästen Bereich hineingeschlagen – vgl. (1) und (2) in Abbildung 5.2. Die Ausfräsung und künstliche Pfanne müssen dabei so zusammenpassen, dass Letztere beim Hineinschlagen fest (pressfit) sitzt [Lür17].

Danach wird bei der Schaftpräparation der schwammartige Markraum des Oberschenkelknochens (Femur) mit Hammer und Meißel eröffnet. Dies geschieht in der sogenannten Viererposition.

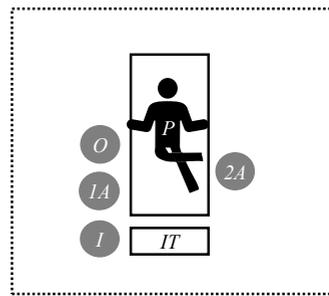
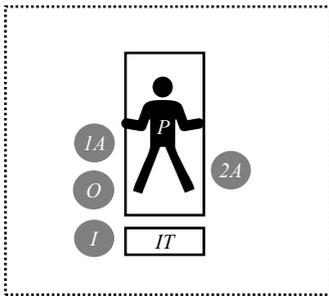
Abbildung 5.3 zeigt eine Momentaufnahme, bei dem der Patient in der Viererposition liegt, also durch die rechtwinklige Positionierung des Unterschenkels des betroffenen Beines auf dem Oberschenkel des anderen Beines der Eindruck einer Vier entsteht. In Folge dieses Vorgehens sind bei dem Eingriff typische Positionswechsel der beteiligten Akteure zu beobachten.

Dann wird der Markraum schrittweise, mit Raffeln (Raspeln) steigenden Durchmessers, freipräpariert (2). Ist die richtige Größe und Tiefe erreicht, verbleibt die letzte Raffel probeweise im Markraum (3) – sie wird mit einem künstlichen Hüftkopf versehen. Zur Probe wird dann das künstliche Gelenk eingerenkt, also der Hüftkopf und die Pfanne reponiert (5). Durch kontrollierte Bewegungen des Beines wird geprüft, ob die Bewegungsausmaße stabil

Tabelle 5.1: Zwei Momentaufnahmen eines prototypischen Operationsssaales bei einer endoprotetischen Versorgung mit einer Hüft-TEP. Der Patient *P* liegt auf dem Rücken auf dem Operationstisch. Operateur *O*, erster Assistent *1A*, zweiter Assistent *2A* und der Instrumentierende *I* stehen um den Tisch herum. Der Instrumentierende *I* reicht die Instrumente von Instrumentiertisch *IT* an die Kollegen weiter. In Anlehnung an [Phi16c, Sch16].

Operationssaal Sicht A

Operationssaal Sicht B



sind und letztlich ob von identischen Beinlängen ausgegangen werden kann [Lür17].

Schließlich wird die Raffel durch den Originalschaft mit Originalkopf ersetzt. Die Wunde am Bein wird danach verschlossen und ein Verband angelegt.

Tabelle 5.1 zeigt zwei Skizzen eines prototypischen Operationsssaales, in welchem die Operation stattfindet. Der Patient *P* liegt auf dem Rücken auf dem Operationstisch.

Der Operateur *O* steht zu Beginn der Operation am Fußende des Tisches, flankiert vom ersten Assistenten *1A* und dem Instrumentierenden *I*. Letzterer reicht die Instrumente vom Instrumentiertisch *IT* an seine Kollegen weiter.

Auf der gegenüberliegenden Seite des Tisches befindet sich der zweite Assistent 2A, der während der Operation vorwiegend Haltearbeit verrichtet sowie bei der Positionierung des Patienten unterstützt (vgl. Abbildung 5.3).

Aufsicht A der Tabelle 5.1 zeigt die Positionierung der Ärzte bis zur Pfannenpräparation – danach, mit Beginn der Schaftpräparation, wird der Patient erstmals in Viererposition verbracht. Operateur O und erster Assistent 1A tauschen daraufhin die Plätze – somit hat O einen direkten Blick auf den Schaft.

Anders als im Falle der minimalinvasiven Chirurgie (vgl. Kapitel 5.1.2), kann bei der technischen Erfassung des Operationsgeschehens der Hüft-TEP nach Bauer nicht auf vorhandene Sensorik zurückgegriffen werden, die bereits Bestandteil der Versorgung ist (beispielsweise Videoendoskop).

Eine Möglichkeit zur Erfassung von Sensorwerten und daraus abgeleiteter Merkmale ist die Verwendung von Tiefenkameras, die an der Decke des Operationssaales angebracht sind und Ärzte, Patient sowie Instrumente aus verschiedenen Perspektiven erfassen können [Phi16c, Sch16].

5.1.2 Laparoskopische Cholezystektomie

Eine laparoskopische Cholezystektomie ist ein chirurgisches Verfahren zur Entfernung (Ektomie)¹ der Gallenblase (engl. Cholecyst). Ausgehend vom Jahr 1882, in dem die erste Cholezystektomie noch am großflächig eröffneten Körper eines Menschen durchgeführt wurde [Tra76, Van86], hat sich die sogenannte laparoskopische Cholezystektomie als Goldstandard der Gegenwart

¹ Hinweis: Im deutschen Sprachraum wird zwischen den Begriffen Ektomie und Resektion unterschieden. Ersteres bezeichnet ein (komplettes) Herausschneiden eines Organs, während Letzteres eher im Sinne des Wegschneidens von Teilen oder aber auch ganzer Organe gebraucht wird. Der Übergang von Resektion zu Ektomie ist also in gewissem Sinne fließend. In dieser Ausarbeitung werden, unter dem Gesichtspunkt der medizinischen Anwendungsfälle, der Begriff Ektomie als Entfernung und der Begriff Resektion als Abschneiden umschrieben.

entwickelt [McS89, Wöl05]. Bei diesem Verfahren wird mit einer minimal-invasiven¹ Zugangsart ein Laparoskop, ein spezielles Endoskop zur Durchführung einer Laparoskopie (= Bauchspiegelung), in den Körper des Patienten eingeführt.

Ein typischer Grund für die Durchführung des Eingriffs sind Gallensteine, die bei den Patienten relevante Schmerzen verursachen [Pér94]. Durch den Eingriff sollen diese Gallensteine einerseits entfernt werden, andererseits wird durch die Entnahme der gesamten Gallenblase die Neubildung weiterer Gallensteine verhindert.

Die Cholezystektomie ist mit leicht rückläufigen Fallzahlen ein vergleichsweise häufiger Eingriff an deutschen Krankenhäusern. Im Jahr 2016 wurden 213.821 Eingriffe vorgenommen, im Jahr 2018 sank die Zahl leicht auf 198.942 Operationen [Sta19b].

In 2018 war die Cholezystektomie damit auf Rang 11 der häufigsten Operation in Deutschland [Sta19b]. Werden solcherlei Eingriffe geeignet technisch unterstützt (vgl. Kapitel 5), so ist zu erwarten, dass dies einen Einfluss auf eine Vielzahl an Patienten und ärztlichem Personal haben wird.

Der Soll-Operationsverlauf der Cholezystektomie kann wegen der gut dokumentierten und standardisierten Vorgehensweise [Sta14] zwischen Personen und Krankenhäusern übertragen werden.

Ähnlich wie bei der Thematik der medizinischen Leitlinien [Phi15b, Phi15c, Phi16a, Phi16b] sollen die Vorgehensweisen aber durch Experten entsprechend den Rahmenbedingungen (beispielsweise Ausstattung des betreffenden Krankenhauses) angepasst und fallspezifisch interpretiert werden.

Abbildung 5.4 zeigt skizzenhaft und stark vereinfacht die anatomische Lage der Gallenblase, der Leber sowie des Magens. Für den Eingriff wird zunächst die Bauchhöhle des Patienten mit Kohlendioxid gefüllt (Insufflation) – diese Ansammlung von Gasen in der Bauchhöhle wird auch als Pneumoperitoneum bezeichnet.

¹ lat. *invadere* = eindringen

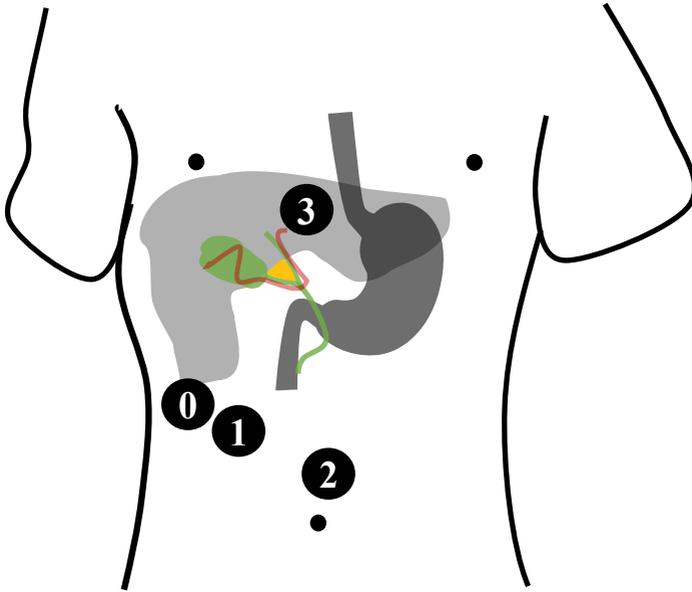


Abbildung 5.4: Skizzenhafte Darstellung eines Oberkörpers mit Leber (hellgrau), Magen (dunkelgrau), Calot'sches Dreieck (gelb) und der Galle mitsamt Gallenwegen (grün) sowie arterieller Verbindung (rot). Im Rahmen einer laparoskopischen Cholezystektomie werden üblicherweise vier künstliche Körperöffnungen geschaffen um die Gallenblase zu entfernen. Die Abbildung zeigt beispielhaft die Lage dieser vier Zugänge (0–3). Die Farben dienen zur Verdeutlichung der Zusammenhänge und sind nicht physiologisch motiviert. In Anlehnung an [Psc13, Tre91, Pad10].

Nach initialer Schaffung des Pneumoperitoneums werden vier künstliche Körperöffnungen geschaffen (vgl. Abbildung 5.4). Dazu wird jeweils ein spitzer Dorn (= Trokar) mit umgebender Hülse verwendet [Psc13, Feu91]. Die vier Trokare werden so positioniert, dass das notwendige Instrumentarium sinnvoll eingeführt werden.

Während des Eingriffs wird zunächst der Gallengang bzw. das Calot'sches Dreieck freipräpariert. Danach wird dieser mit Clips verschlossen (sogenanntes Clippen) und schließlich schneidend durchtrennt. Ebenso wird mit dem

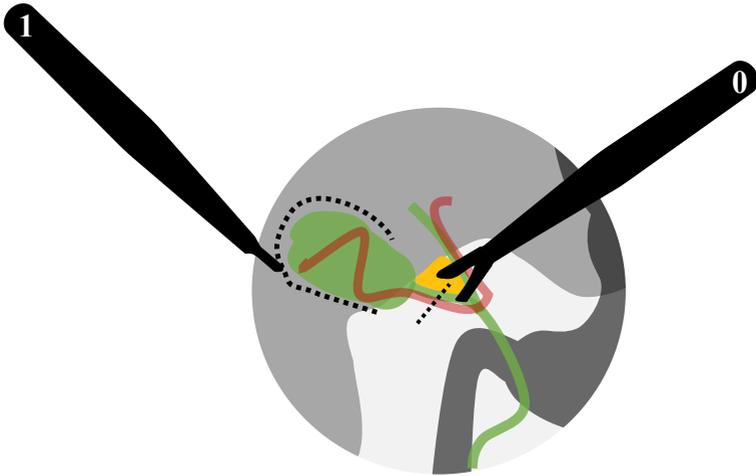


Abbildung 5.5: Die Abbildung zeigt skizzenhaft die bei der Cholezystektomie notwendige Durchtrennung (0) des Gallenweges (grün) und der arteriellen Blutversorgung (rot). Weiterhin ist das sogenannte Calot'sche Dreieck (gelb) dargestellt, welches als Orientierungshilfe im Bereich Leber, Gallenwege und Arterien dient. Im späteren Verlauf des Eingriffs wird die Gallenblase vom Leberbett entfernt (1). In Anlehnung an [Feu91].

Gefäß verfahren, welches die Gallenblase mit Blut versorgt (vgl. Abbildung 5.5). Optional wird während des Eingriffs ein bildgebendes Verfahren eingesetzt, ein sogenannte intraoperative Cholangiographie. Danach wird die Gallenblase vom Leberbett entfernt.

Letzteres wird dann zur Blutstillung mittels Elektrokoagulation, also einer Zerstörung von Gewebe mittels Strom, versorgt. Abbildung 5.6 zeigt einen

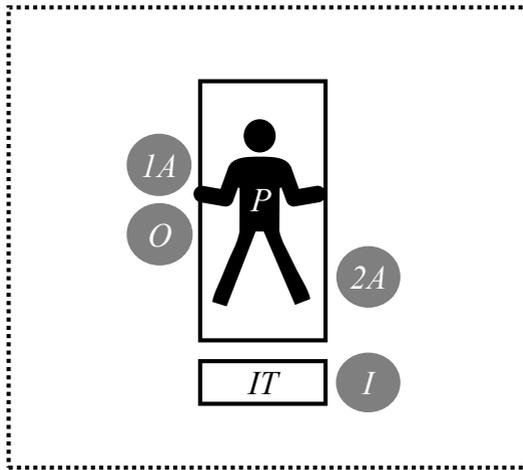


Abbildung 5.6: Die Abbildung zeigt eine Momentaufnahme eines prototypischen Operationszimmers bei einer laparoskopischen Cholezystektomie. Der Patient *P* liegt rücklings auf dem Operationstisch. Operateur *O*, erster Assistent *1A*, zweiter Assistent *2A* und der Instrumentierende *I* stehen um den Tisch herum. Der Instrumentierende *I* reicht die Instrumente von Instrumentiertisch *IT* weiter. Weitere Personen, wie beispielsweise der Anästhesist oder der Springer sind zur Vereinfachung nicht Teil der Abbildung. In Anlehnung an [Feu91].

prototypischen Operationsaal, in welchem die Operation stattfindet¹. Der Patient *P* befindet sich rücklings auf dem Operationstisch. Operateur *O* und zweiter Assistent *2A* befinden sich auf der rechten Flanke des Patienten, während der erste Assistent auf der gegenüberliegenden Seite positioniert ist. Der Instrumentierende *I* steht am Fußende und reicht bei Bedarf das notwendige Instrumentarium an oder nimmt dieses wieder entgegen² [Feu91, Pad10].

¹ Hinweis: Der Operationsaal selbst ist in den sogenannten OP eingebettet, der üblicherweise über Eingangs- und Ausgangsschleusen für Personal und Patienten betreten werden kann. Der OP kann darüber hinaus weitere Operationssäle, sowie Aufenthalts- und Lagerräume umfassen.

² Hinweis: In der Literatur wird auch von Instrumentierschwester, OP-Schwester. Schwester oder Instrumentierende gesprochen.

5.2 Konstituenten des Problems

In den folgenden Kapiteln werden zunächst die Begrifflichkeiten zur Beschreibung der Problemstellung eingeführt (Kapitel 5.2.1) – daran anschließend werden mögliche Bewertungen der Problemlösung besprochen (Kapitel 5.2.2).

5.2.1 Begrifflichkeiten

Abbildung 5.7 zeigt ein Arbeitsablaufmodell der endoprothetischen Versorgung eines Patienten mit einer Hüft-TEP anhand eines verkürzt dargestellten UML-Aktivitätsdiagramms, vgl. Kapitel 3.3 und 5.1.1. Dieser Formalismus wurde bereits in früheren Forschungsarbeiten zur Modellierung medizinischer Arbeitsabläufe [Phi15b, Phi15c, Phi16a, Phi16b, Phi17d] verwendet.

Der Start der Aktivität Hüft-TEP wird durch einen ausgefüllten Kreis (Startknoten) dargestellt, während der Doppelkreis (Endknoten) das Ende der Aktivität repräsentiert. Durchgeführte Aktionen werden in dieser Darstellung als abgerundete Rechtecke symbolisiert, gerichtete Kanten (Pfeile) stellen den Kontrollfluss dar. Im Beispiel werden die Aktionen sequentiell – d.h. also nacheinander ausgeführt, vgl. Kapitel 3.3.2.

Ziel ist es nun, den vormodellierten medizinischen Arbeitsablauf mit den aktuellen Gegebenheiten im Operationssaal abzugleichen – und somit den Ist-Fortschritt der Operation zu bestimmen. Von Interesse ist es also beispielsweise, ob gerade der Zugangsweg freipräpariert oder aber etwa der Hautschnitt durchgeführt wird (vgl. Abbildung 5.7).

Auf der anderen Seite existieren über eine Sensierung der Gegebenheiten im Operationssaal und nachfolgender Merkmalsbildung entsprechende Merkmalswerte über der Zeit. Für den gegebenen Anwendungsfall können Merkmale beispielsweise die Anzahl der Personen am Operationstisch oder die Anzahl verwendeter Instrumente sein.

Merkmale bzw. deren Instanziierungen in Form von konkreten Merkmalswerten können durch einen Merkmalsvektor \mathbf{x} gebündelt werden. Dieser wird im

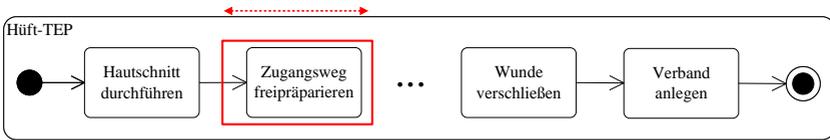


Abbildung 5.7: Die Abbildung zeigt ein vereinfachtes Modell des medizinischen Arbeitsablaufs einer endoprothetischen Versorgung eines Patienten mit einer Hüft-TEP. Ziel ist es, den vormodellierten Ablauf mit den aktuellen Gegebenheiten im Operationssaal abzugleichen um dadurch den Ist-Fortschritt der Operation schätzen zu können. Als visuelle Hilfsmittel sind der Abbildung des Modells ein rotes Rechteck, ein roter Pfeil sowie drei Punkte (...) hinzugefügt – diese sind nicht Teil der UML-Spezifikation [OMG17]. Die drei Punkte symbolisieren eine Abfolge beliebiger Aktionen, die der Einfachheit halber nicht dargestellt sind. Das rote Rechteck und der Pfeil symbolisieren den geschätzten Ist-Fortschritt, der während der Operation sukzessive, in Abhängigkeit der stattfindenden Beobachtungen, angepasst wird.

Folgenden als Spaltenvektor definiert, dessen Elemente gerade die einzelnen Merkmalswerte sind. Also:

$$\mathbf{x} := (x_0, \dots, x_{d-1})^T. \tag{5.1}$$

Dabei repräsentiert der Skalar d die Anzahl der gebündelten Merkmale und damit die Dimension des resultierenden Merkmalsraumes welches dem technischen System zur Lösung des Problems zur Verfügung steht. Um Platz zu sparen werden Merkmalsvektoren \mathbf{x} in dieser Ausarbeitung transponiert, also horizontal, dargestellt und in Folge mit dem Symbol eines hochgestellten T versehen¹.

In jedem Zeitschritt wird ein neuer Merkmalsvektor \mathbf{x} mit Index t seitens des technischen Systems im Operationssaal beobachtet. Dieser wird in dieser Ausarbeitung festgelegt als:

$$\mathbf{x}^t := (x_0^t, \dots, x_{d-1}^t)^T. \tag{5.2}$$

¹ Hinweis: Eine Auflistung aller Symbole findet sich im Symbolverzeichnis ab Seite 209.

Infolgedessen ergibt sich über die Zeit eine Beobachtungssequenz von Merkmalsvektoren

$$\mathbf{x}^{0:t} := \mathbf{x}^0, \dots, \mathbf{x}^t. \quad (5.3)$$

Die Beobachtungssequenz $\mathbf{x}^{0:t}$ steht dem technischen System als Beobachter der Gegebenheiten im Operationssaal zur Verfügung wohingegen der aktuelle Ist-Fortschritt (vgl. Abbildung 5.7, rote Markierung) nicht direkt beobachtbar ist und somit verborgen bleibt. Das bedeutet, es besteht eine Lücke zwischen der Beobachtungssequenz $\mathbf{x}^{0:t}$ auf der einen und dem zu ermittelnden Ist-Fortschritt innerhalb des Modells auf der anderen Seite (o.B.d.A. vgl. Abbildung 5.8). Diese Lücke soll im Anwendungsfall zur Laufzeit (engl. online), unter Kenntnis der bisher beobachteten Merkmalsvektoren aber ohne Kenntnis der noch zukünftig kommenden – also sprich ohne Kenntnis der gesamten Beobachtungssequenz (engl. offline) – geschlossen werden.

In traditionellen Ansätzen wie den Hidden Markov Modellen (HMM) werden hierzu Wahrscheinlichkeitsverteilungen verwendet, um anzugeben, wie wahrscheinlich es ist, dass ein bestimmter Merkmalsvektor \mathbf{x} in einer bestimmten Phase beobachtet wird. In anderen Worten: Man nutzt die bedingte Wahrscheinlichkeit für das Vorliegen eines bestimmten Merkmalsvektors \mathbf{x} gegeben die Phase (vgl. Abbildung 5.8, untere Zeile).

Unter Verwendung weiterer Parameter des HMM kann umgekehrt die Wahrscheinlichkeit geschätzt werden, dass gegebenen eine Beobachtungssequenz eine bestimmte Phase vorliegt. Dadurch wird die Lücke zwischen der Beobachtungssequenz und des Ist-Fortschritts im Modell überbrückt (vgl. Abbildung 5.8, oberer Bereich).

Bezogen auf das zugrunde liegende Klassifikationsproblem, stellen die einzelnen Phasen die Klassen $\omega \in \Omega$ dar. Durch Lernstichproben mit Merkmalsvektoren bekannter Klassenzugehörigkeiten $\omega(\mathbf{x})$, können die Klassifikationsmodelle überwacht eingelernt werden. Zur nachfolgenden Klassifikation von Merkmalsvektoren mit unbekannter Klassenzugehörigkeit kommen dann die modellspezifischen Ansätze zur Schätzung der Klassenzugehörigkeit

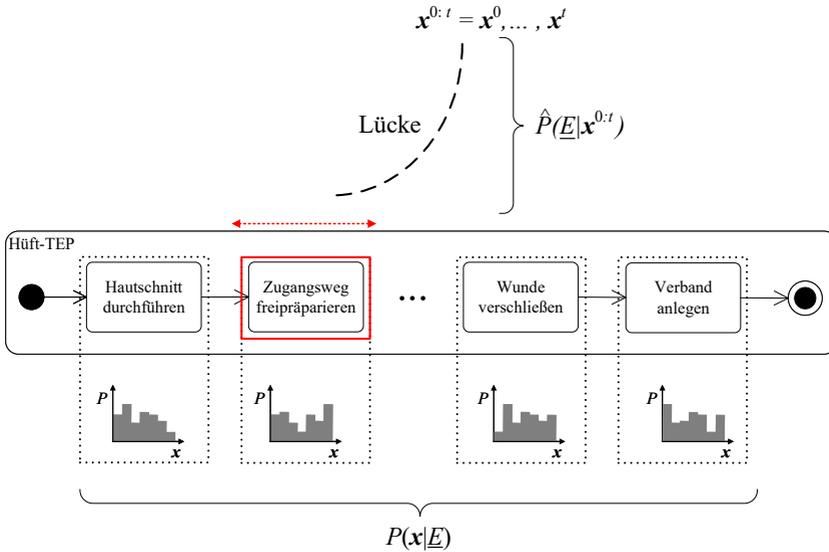


Abbildung 5.8: Die Abbildung zeigt die Lücke zwischen einer Sequenz von Merkmalsvektoren und dem Ist-Fortschritt des Modells des Arbeitsablaufes. Letzterer ist dem technischen System (o.B.d.A.) verborgen. Um die Lücke dazwischen zu überbrücken, kann mit Hilfe einer Wahrscheinlichkeitsverteilung angegeben werden, wie wahrscheinlich es ist, dass ein bestimmter Merkmalsvektor in einer bestimmten Phase beobachtet wird – spricht: $P(\underline{x}|\underline{E})$.

$\hat{P}(\omega|\underline{x})$ zum Tragen – diese und weitere Einzelheiten der Modellcharakteristika werden in den folgenden Abschnitten, unter Berücksichtigung des Standes der Technik, näher besprochen.

Die beschriebenen Begrifflichkeiten sollen nun noch in Bezug zum Begriff des Expertensystems, vgl. Kapitel 2.2, gesetzt werden. Über die Problemstellung liegt einerseits bereichsspezifisches Wissen vor, beispielsweise über typische Operationsabläufe oder typischerweise verwendete Werkzeuge in den einzelnen OP-Phasen. Das bereichsspezifische Wissen bildet gemeinsam mit dem fallspezifischen Wissen die Wissensbasis des Expertensystems. Während das

bereichsspezifische Wissen gemeinsam mit Experten formalisiert und datengetrieben erweitert wird, wird das fallspezifische Wissen in der hier vorliegenden Problemstellung ohne Beteiligung des Experten, allein durch Sensierung erhoben. Es liegt damit nach [Pup13a] ein eingebettetes Expertensystem vor, vgl. Kapitel 2.2, in welchem das fallbasierte Wissen durch die Beobachtungssequenz $\mathbf{x}^{0:t}$ gegeben ist. Das fallbasierte Wissen liegt somit online vor, wird nach und nach aufgebaut und nach der Bearbeitung eines spezifischen Anwendungsfalls wieder gelöscht.

5.2.2 Bewertung der Problemlösungen

In der Literatur finden sich verschiedene Maße zur Bewertung der Lösung des in Kapitel 5.2.1 beschriebenen Problems. Hierbei muss letztlich die Klassenzugehörigkeit von Merkmalsvektoren geschätzt werden, um die Lücke zwischen subsymbolischen Merkmalsausprägungen und den auf symbolischer Ebene gelegenen medizinischen Phasen, also den Klassen, zu schließen. Wie häufig nun eine Phase korrekt geschätzt wurde, also wie häufig bei der Problemlösung der Idealfall $\hat{\omega}(\mathbf{x}) = \omega(\mathbf{x})$ vorlag, kann anschaulich mit einer sogenannten Konfusionsmatrix visualisiert werden.

Mittels einer solchen Matrix werden die wahren Klassenzugehörigkeiten $\omega(\mathbf{x})$ den durch die Klassifikation geschätzten Klassenzugehörigkeiten $\hat{\omega}(\mathbf{x})$ gegenüber gestellt. Für den Fall zweier Klassen ergibt sich eine Matrix \mathbf{C} mit den in Abbildung 5.9 dargestellten, charakteristischen Einträgen $c_{i,j}$ mit $i, j \in [0,1]$.

Abgeleitet von der üblichen Einteilung der beiden involvierten Klassen in eine sogenannte Positiv- sowie Negativklasse repräsentieren die true positives (tp) und true negatives (tn) die Anzahl der Fälle, in denen bei der Problemlösung die geschätzte Klassenzugehörigkeit jeweils mit der wahren Klassenzugehörigkeit der Merkmalsvektoren übereinstimmte. Sprich: die Diagonalelemente $c_{0,0}$ und $c_{1,1}$ der Matrix \mathbf{C} bündeln die Anzahl der Fälle, in denen Merkmalsvektoren gemäß ihrer Klassenzugehörigkeit korrekt klassifiziert wurden, sich also die geschätzte Klassenzugehörigkeit als tatsächlich zutreffend, also als wahr (engl. true), herausgestellt hat.

		$\hat{\omega}(\mathbf{x})$	
		pos.	neg.
$\omega(\mathbf{x})$	pos.	tp	fn
	neg.	fp	tn

Abbildung 5.9: Konfusionsmatrix für den Zweiklassenfall. Die true positives (tp) und true negatives (tn) repräsentieren die Anzahl der Fälle, in denen bei der Problemlösung die geschätzte Klassenzugehörigkeit jeweils mit der wahren Klassenzugehörigkeit der Merkmalsvektoren übereinstimmt. Zum Anderen können Merkmalsvektoren irrtümlich als positiv oder aber negativ klassifiziert werden – in diesen Fällen war die Schätzung falsch positiv oder aber falsch negativ. Die Anzahl dieser Fälle finden sich als fp (falsch positiv) oder aber als fn (falsch negativ) in den Matrixeinträgen abseits der Diagonale wieder.

Von diesen Idealfällen kann es allerdings Abweichungen geben – nämlich dann, wenn die Schätzung der Klassenzugehörigkeit von Merkmalsvektoren eben nicht mit deren wahren Klassenzugehörigkeit übereinstimmt. Im Zweiklassenfall gibt es für diese Konfusion, welche sich abseits der Diagonalelemente abspielt, zwei Möglichkeiten (vgl. Abbildung 5.9).

Zum Einen kann eine in Wahrheit vorliegende Zugehörigkeit zur Positivklasse als eine Zugehörigkeit zur Negativklasse geschätzt worden sein, sprich: Merkmalsvektoren die eigentlich zur Positivklasse zugehörig sind, wurden fälschlicherweise negativ klassifiziert – diese Schätzungen sind somit falsch negativ (engl. false negative).

Zum Anderen können Merkmalsvektoren in entsprechender Weise irrtümlich als positiv klassifiziert worden sein – in diesen Fällen war die Schätzung falsch positiv (engl. false positive). Die Anzahl der beschriebenen Fälle wird über die Matrixeinträge $c_{0,1}$ für die false negatives (fn) und $c_{1,0}$ für die false positives (fp) zusammengefasst.

Aus dieser Einteilung der Klassifikationsergebnisse lassen sich verschiedene Kennzahlen gewinnen, die zur Quantifizierung der Qualität der Problemlösung aus verschiedenen Perspektiven herangezogen werden können. Eine solche Kennzahl ist die Genauigkeit (engl. Accuracy¹):

$$\text{Genauigkeit} = \frac{tp + tn}{tp + tn + fp + fn} \in [0,1]. \quad (5.4)$$

Hier werden die tatsächlichen Treffer der Diagonalen, bei denen also $\omega(\mathbf{x}) = \hat{\omega}(\mathbf{x})$ gilt, ins Verhältnis zur Summe aller Schätzungen gesetzt² (vgl. Matrix \mathbf{C} in Abbildung 5.9).

Die sogenannte Precision (auch: Positive Predictive Value, PPV oder Relevanz) stellt die Anzahl der als korrekt zur Positivklasse gehörig klassifizierten Merkmalsvektoren ins Verhältnis zur Anzahl derjenigen Fälle, in denen der Klassifikator ein Vorliegen der Positivklasse geschätzt hat:

$$\text{Precision} = \frac{tp}{tp + fp} \in [0,1]. \quad (5.5)$$

Irrt der Klassifikator bezüglich des Vorliegens der Positivklasse nie, es gilt also $fp = 0$, ergibt sich die bestmögliche Precision mit dem Wert 1. Dieser Wert ist aber letztlich nur eine Seite der Medaille.

Mit dem sogenannten Recall (auch: True Positive Rate, TPR oder Sensitivität) können umgekehrt die Anzahl der als korrekt zur Positivklasse gehörig klassifizierten Merkmalsvektoren ins Verhältnis zu den tatsächlich vorhandenen

¹ Hinweis: Auch finden sich in der Literatur die ins Deutsche übersetzten Begriffe Akkuratheit, Akkuranz und Akkuratesse.

² Hinweis: Der Begriff der Genauigkeit findet sich in verschiedenen Domänen mit andersartiger Bedeutungen. In dieser Ausarbeitung wird der Begriff stets im Kontext der Klassifikation verwendet und ist entsprechend der oben genannten Definition in Gleichung (5.4) zu verstehen.

Positivfällen in der Stichprobe gesetzt werden:

$$Recall = \frac{tp}{tp + fn} \in [0,1]. \quad (5.6)$$

Entsprechend ergibt sich der Bestwert 1 des Recalls in dem Falle, in dem keinerlei Merkmalsvektoren der Positivklasse übersehen wurden. In einem solchen Fall gilt $fn = 0$.

Zur Verschmelzung beider Aspekte wird üblicherweise das harmonische Mittel¹ aus Recall (5.6) und Precision (5.5) gebildet:

$$F_1\text{-Score} = \frac{2}{Recall^{-1} + Precision^{-1}} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (5.7)$$

$$= \frac{2 \cdot tp}{2 \cdot tp + fp + fn} = \frac{tp}{tp + \frac{1}{2} \cdot (fp + fn)} \in [0,1]. \quad (5.8)$$

Dabei werden fp und fn in gleicher Weise berücksichtigt (5.8). Neben dem sogenannten F_1 -Score – häufig auch kurz F-Score genannt – gibt es den allgemeineren F_β -Score der verschiedene Gewichtungen beider Aspekte ermöglicht:

$$F_\beta\text{-Score} = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \quad (5.9)$$

$$= \frac{(1 + \beta^2) \cdot tp}{(1 + \beta^2) \cdot tp + fp + \beta^2 \cdot fn} \quad (5.10)$$

$$= \frac{tp}{tp + \frac{1}{(1+\beta^2)} \cdot (fp + \beta^2 \cdot fn)} \in [0,1]. \quad (5.11)$$

¹ Hinweis: $\bar{x}_{harm} = \frac{n}{x_1^{-1} + \dots + x_n^{-1}}$.

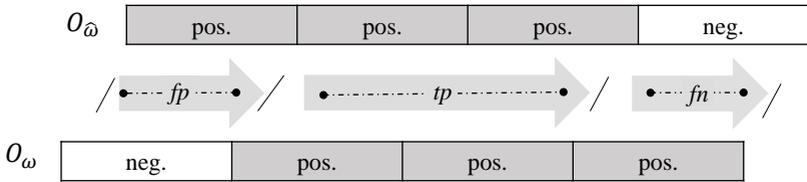


Abbildung 5.10: Im hinteren, oberen Bereich ist einerseits mit $O_{\hat{\omega}}$ eine beispielhafte konkrete Ausprägung der geschätzten Klassenzugehörigkeiten $\hat{\omega}(\mathbf{x})$ über die Zeit dargestellt. Im vorderen, unteren Bereich ist mit O_{ω} andererseits eine beispielhafte konkrete Ausprägung der tatsächlichen, wahren Klassenzugehörigkeiten $\omega(\mathbf{x})$ über die Zeit skizziert. Dort, wo gedanklich $O_{\hat{\omega}}$ und O_{ω} auf der gemeinsamen zeitlichen Sequenz zum Schnitt kommen, liegen die tp vor. Sprich die geschätzte Klassenzugehörigkeit stimmt mit der wahren Klassenzugehörigkeit überein. Andere Bereiche, die also nicht im Schnitt liegen, sind entweder als fp oder fn einzustufen und auf der Abbildung entsprechend markiert.

Für die gegebene Problemstellung (vgl. Kapitel 5.2.1) wird allerdings statt des F-Scores üblicherweise der Jaccard-Score¹ herangezogen. Dieser ist in diesem Zusammenhang definiert als:

$$\text{Jaccard-Score} = \frac{tp}{tp + fn + fp} \in [0,1]. \quad (5.12)$$

Der Kennwert lässt sich zurückführen auf den sogenannten Jaccard-Koeffizienten [Jac12] (auch: Jaccard-Index), bei dem der Quotient aus dem Betrag des Schnitts (engl. Intersection) und dem Betrag der Vereinigung (engl. Union) zweier Mengen gebildet wird. Deswegen ist der Jaccard-Koeffizient auch unter dem Begriff Intersection over Union (IoU) bekannt².

Im vorliegenden Falle werden die geschätzte Klassenzugehörigkeit $\hat{\omega}(\mathbf{x})$ und die tatsächliche Klassenzugehörigkeit $\omega(\mathbf{x})$ der Sequenz von Merkmalsvektoren verglichen. Dort wo beide übereinstimmen, also im Schnitt, liegen dann

¹ Hinweis Häufig wird der Jaccard-Score verkürzt als Jaccard bezeichnet.

² Hinweis: Der IoU spielt in der Objekterkennung eine wichtige Rolle – dort wird der zu erkennende Bildbereich mit dem tatsächlich erkannten Bildbereich über die Schnitt- bzw. Vereinigungsmenge in Beziehung gesetzt.

die tp vor. Abseits davon wurden entweder Merkmalsvektoren der Positivklasse übersehen (fn) oder dieser irrtümlich zugewiesen (fp).

Seien beispielsweise die geschätzten und die tatsächlichen Klassenzugehörigkeiten vierer Merkmalsvektoren $\mathbf{x}_0, \dots, \mathbf{x}_3$ beschrieben durch:

$$O_\omega = (\omega(\mathbf{x}^0), \omega(\mathbf{x}^1), \omega(\mathbf{x}^2), \omega(\mathbf{x}^3)),$$

$$O_{\hat{\omega}} = (\hat{\omega}(\mathbf{x}^0), \hat{\omega}(\mathbf{x}^1), \hat{\omega}(\mathbf{x}^2), \hat{\omega}(\mathbf{x}^3)).$$

Dann ergibt sich für eine konkrete Ausprägung

$$O_\omega = (\text{pos.}, \text{pos.}, \text{pos.}, \text{neg.}),$$

$$O_{\hat{\omega}} = (\text{neg.}, \text{pos.}, \text{pos.}, \text{pos.}),$$

der in Abbildung 5.10 dargestellte Sachverhalt. Überlappungen bezüglich der geschätzten und der tatsächlichen Klassenzugehörigkeit ergeben sich als Schnittmenge, wenn O_ω und $O_{\hat{\omega}}$ gedanklich als Mengen auf die Sequenz von Merkmalsvektoren projiziert werden.

In ähnlicher Weise lassen sich Precision und Recall interpretieren; Während im Zähler der Betrag der Schnittmenge erhalten bleibt, finden sich in den Nennern der Kennwerte entweder der Betrag der Menge der Schätzungen (Precision) oder der Betrag der Menge der tatsächlichen Klassenzugehörigkeiten (Recall).

Wie bereits eingangs erwähnt, lassen sich die Klassifikationsergebnisse übersichtlich in einer Konfusionsmatrix darstellen (vgl. Abbildung 5.9). Ein Zahlenbeispiel soll dies im Folgenden verdeutlichen.

		$\hat{\omega}(x)$	
		pos.	neg.
$\omega(x)$	pos.	5	4
	neg.	0	1

		$\hat{\omega}(x)$		
		pos.	neg.	
$\omega(x)$	pos.	<i>tp</i>	<i>fn</i>	<i>Recall</i>
	neg.	<i>fp</i>	<i>tn</i>	

Precision

Abbildung 5.11: Links: Konfusionsmatrix für den Zweiklassenfall mit konkreten Ausprägungen der Einträge aus Abbildung 5.9. Rechts: Konfusionsmatrix mit skizzenhafter Markierung der betroffenen Einträge bei Berechnung von Precision und Recall.

Für die genannten Kennzahlen ergeben sich am Beispiel der in Abbildung 5.11 dargestellten Konfusionsmatrix:

$$Precision = \frac{tp}{tp + fp} = \frac{5}{5 + 0} = 1,$$

$$Recall = \frac{tp}{tp + fn} = \frac{5}{5 + 4} = \frac{5}{9},$$

$$F_1\text{-Score} = \frac{tp}{tp + \frac{1}{2} \cdot (fp + fn)} = \frac{5}{5 + \frac{1}{2} \cdot (0 + 4)} = \frac{5}{7},$$

$$Jaccard = \frac{tp}{tp + fp + fn} = \frac{5}{5 + 0 + 4} = \frac{5}{9},$$

$$Genauigkeit = \frac{tp + tn}{tp + tn + fp + fn} = \frac{5 + 1}{5 + 1 + 0 + 4} = \frac{6}{10}.$$

Dabei zeigt die Precision mit dem Maximalwert 1 an, dass der Klassifikator bezüglich des Vorliegens der Positivklasse nie irrt und somit die Anzahl der *fp* bei 0 liegt.

Bei Recall indes, wurden 4 Merkmalsvektoren der Positivklasse „übersehen“ und somit irrtümlich der Negativklasse zugeordnet. Da gleichzeitig 5 Merkmalsvektoren korrekt zugeordnet wurden, ergibt sich in 5 von insgesamt 9

Fällen eine korrekte Zuordnung der Merkmalsvektoren der Positivklasse. Der resultierende Recall ergibt sich zu $\frac{5}{9}$.

Der F_1 -Score bezieht beide Arten der Fehlklassifikation in gleicher Weise ein, indem fp und fn arithmetisch gemittelt werden. Hierdurch ergibt sich im Beispiel ein Zähler von 7 und der F_1 -Score ist insgesamt gegeben durch $\frac{5}{7}$. Beim Jaccard-Score fließen die Fehlklassifikationen ungemittelt ein – es ergibt sich ein Jaccard-Score von $\frac{5}{9}$. Da im Beispiel $fp = 0$ gilt, entspricht der Jaccard-Score dem Recall.

Betrachtet man alle 10 Klassifikationsversuche in Abbildung 5.11, so lag dort in 6 von 10 Fällen eine korrekte Zuordnung zur jeweiligen Klasse vor (vgl. Diagonalwerte). Die Genauigkeit ergibt sich somit zu $\frac{6}{10}$.

Für den Mehrklassenfall verschmieren sich die fn , fp und tn auf die Einträge der jeweils übrigen Klassen. Die Konfusionsmatrix bläht sich in Zeilen- und Spaltenzahl entsprechend der Klassenanzahl auf. Abbildung 5.12 verdeutlicht dies links für den Fall der geschätzten Klassenzugehörigkeit zur wahren Klasse ω_0 und rechts beispielhaft für den Fall der geschätzten Klassenzugehörigkeit zur wahren Klasse ω_2 .

Werden die Einträge der markierten Flächen für fn , fp und tn pro geschätzter Klassenzugehörigkeit zur wahren Klasse ω_n auf aggregiert ergeben sich n Konfusionsmatrizen gemäß Abbildung 5.9. Hieraus können die klassenweisen Kennwerte gemäß Formeln (5.6), (5.5), (5.7) und (5.4) berechnet werden.

Um die Problemlösung anhand der dargestellten Kennzahlen fundiert bewerten zu können, muss die Stichprobe $\{\mathbf{x}_0, \dots, \mathbf{x}_n\}$ mit bekannten Klassenzugehörigkeiten $\omega(\mathbf{x}_j)$ für $j = 0, \dots, n$ mit Bedacht genutzt werden.

Wird die komplette Stichprobe zum Einlernen des Klassifikators mit nachgeschalteter Bewertung verwendet, kann dessen Fähigkeit zur Generalisierung nicht bewertet werden. Üblicherweise wird deswegen eine Partitionierung der Gesamtstichprobe in eine Lern- und Teststichprobe vorgenommen.¹

¹ Hinweis: Oft wird hierbei auch von Lern- und Testmenge gesprochen.

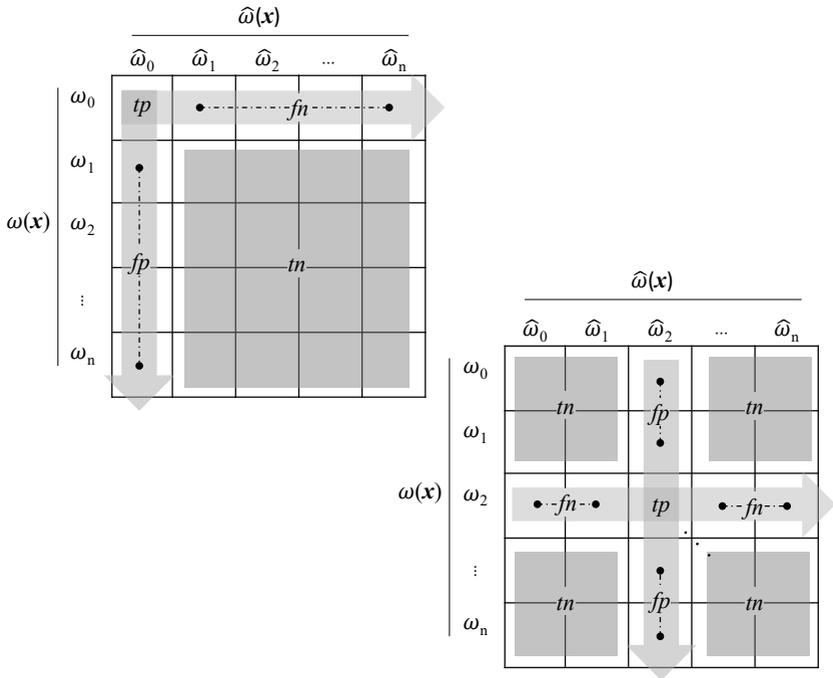


Abbildung 5.12: Konfusionsmatrizen im Falle von n Klassen mit $n > 2$ (Mehrklassenfall). Links: tp , fn , fp und tn bezüglich der geschätzten Klassenzugehörigkeit $\hat{\omega}_0$. Rechts: Darstellung beispielhaft für die geschätzte Klassenzugehörigkeit $\hat{\omega}_2$, welche sich entsprechend für die anderen skizzierten Klassen entlang der Diagonale ergibt.

Sprich:

$$\mathcal{D}_{\text{Gesamt}} = \mathcal{D}_{\text{Lern}} \cup \mathcal{D}_{\text{Test}}, \quad (5.13)$$

so dass gilt:

$$\mathcal{D}_{\text{Lern}} \cap \mathcal{D}_{\text{Test}} = \emptyset. \quad (5.14)$$

Dabei dient $\mathcal{D}_{\text{Lern}}$ zum Einlernen des Klassifikators, wohingegen $\mathcal{D}_{\text{Test}}$ erst nach dem Lernvorgang zur finalen Bewertung der Problemlösung herangezogen wird.

Bei geringen Stichprobengrößen und der damit einhergehenden Problematik der großen Varianz geschätzter Fehlerwahrscheinlichkeiten bzw. nachgelagert der oben genannten Kennwerte, bieten sich Verfahren an, die die vorhandenen Stichproben clever ausnutzen.

Eine übliche Vorgehensweise ist die m -fache Kreuzvalidierung, bei der die Einteilung in Lern- und Teststichprobe rollierend über der Gesamtstichprobe erfolgt. Die Gesamtstichprobe wird zunächst in m annähernd gleich große Partitionen geteilt, d. h.:

$$\mathcal{D}_{\text{Gesamt}} = \{\mathcal{D}_i : i = 0, \dots, m - 1\}, m \geq 2, \quad (5.15)$$

so dass gilt:

$$\mathcal{D}_i \cap \mathcal{D}_j = \emptyset \quad \forall i \neq j, \quad (5.16)$$

$$|\mathcal{D}_i| \approx |\mathcal{D}_j| \quad \forall i \neq j. \quad (5.17)$$

In jedem Durchlauf der Kreuzvalidierung fungiert eine andere Teilmenge \mathcal{D}_i gemäß Gleichungen (5.15) bis (5.17) als Teststichprobe. Die restlichen Teilmengen dienen als Lernstichprobe. In Folge tritt ein Klassifikator in m Durchläufen zum Leistungstest an, wodurch m Kennwerte zur Leistungsmessung zur Verfügung stehen. Üblicherweise werden durch Bildung des arithmetischen Mittels die abschließenden Leistungswerte der Kreuzvalidierung berechnet. Weiterhin lassen sich Varianz und Standardabweichung der Kennwerte bestimmen, was für einen Leistungsvergleich unterschiedlicher Klassifikatoren von Nutzen ist.

Abbildung 5.13 zeigt die beschriebenen Sachverhalte für eine 4-fache Kreuzvalidierung. In den Zeilen sind die Durchläufe 0-3 dargestellt. In Hellgrau ist jeweils jene Teilmenge hervorgehoben, die für einen gegebenen Durchlauf

Durchlauf	Wahl der Teilmengen als Lern- und Teststichprobe			
0	\mathcal{D}_0	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3
1	\mathcal{D}_0	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3
2	\mathcal{D}_0	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3
3	\mathcal{D}_0	\mathcal{D}_1	\mathcal{D}_2	\mathcal{D}_3

Abbildung 5.13: Die Abbildung zeigt eine 4-fache Kreuzvalidierung. In den Zeilen sind die 4 Durchläufe sowie die jeweilige Wahl der Teilmengen als Lernstichprobe (weiß) oder aber als Teststichprobe (hellgrau) dargestellt. In jedem Durchlauf 0-3 wird eine andere Teilmenge \mathcal{D}_i mit $i = 0 \dots 3$ als Teststichprobe verwendet. Im letzten Durchlauf gilt beispielsweise: $\mathcal{D}_{\text{Test}} := \mathcal{D}_3$, sowie $\mathcal{D}_{\text{Lern}} := \mathcal{D}_0 \cup \mathcal{D}_1 \cup \mathcal{D}_2$.

als Testmenge herangezogen wird. Die Vereinigung der restlichen Teilmengen des Durchlaufs wird als Lernstichprobe verwendet. D. h. für Durchlauf 0 gilt: $\mathcal{D}_{\text{Test}} := \mathcal{D}_0$, sowie $\mathcal{D}_{\text{Lern}} := \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$.

Für die abschließenden Leistungswerte, wie beispielsweise den F_1 -Score, erhält man:

$$\overline{F_1\text{-Score}} = \frac{1}{4} \sum_{i=0}^3 F_1\text{-Score}_i, \quad (5.18)$$

$$s_{F_1\text{-Score}} = \sqrt{\frac{1}{3} \sum_{i=0}^3 (F_1\text{-Score}_i - \overline{F_1\text{-Score}})^2}. \quad (5.19)$$

Wobei $F_1\text{-Score}_i$ den im i -ten Durchlauf erzielten F_1 -Score des Klassifikators beschreibt. Wird die Anzahl der Durchläufe maximal, also mit $m = |\mathcal{D}_{\text{Gesamt}}|$ (vgl. Gleichung (5.15)) gewählt, so spricht man von einer „leave-one-out“ Kreuzvalidierung. Wörtlich werden hier also zum Lernen in jedem Durchlauf alle Elemente der Gesamtstichprobe bis auf ein Element genutzt.

Es stellt sich abschließend noch die Frage, welche Modellkonfiguration für die Leistungsmessung mittels Kreuzvalidierung ausgewählt werden soll. Um die möglichen Modellkonfigurationen nicht auf die Teststichproben hin zu optimieren, sollte neben Lern- und Teststichprobe eine weitere Stichprobe etabliert werden. Im Rahmen der eben vorgestellten Kreuzvalidierung bietet es sich an, die dargestellte Vorgehensweise verschachtelt anzuwenden [Dei20].

Dazu wird in einer äußeren Schleife die bereits bekannte rollierende Einteilung in Test- und Lernstichproben vorgenommen. Darüber hinaus wird zusätzlich innerhalb der angedachten Lernstichproben eine weitere rollierende Einteilung in Lern- und sogenannte Validierungsstichproben umgesetzt. In der letztgenannten, inneren Schleife werden verschiedene Modellkonfigurationen bewertet. Die abschließende Leistungsmessung erfolgt mit der noch unbekanntem Teststichprobe der äußeren Schleife.

Abbildung 5.14 zeigt das Vorgehen für eine 4-fache Kreuzvalidierung, in welche eine 3-fache Kreuzvalidierung eingebettet ist. Im äußeren Durchlauf 0 gilt für die Teststichprobe (hellgrau): $\mathcal{D}_{\text{Test}} := \mathcal{D}_0$. Die Vereinigung der restlichen Mengen wird einer weiteren rollierenden Einteilung in Lernstichproben (weiß) sowie Validierungsstichproben (dunkelgrau) zugeführt. Für Durchlauf 0,0 gilt somit: $\mathcal{D}_{\text{Val}} := \mathcal{D}_1$ und $\mathcal{D}_{\text{Lern}} := \mathcal{D}_2 \cup \mathcal{D}_3$.

Bei den vorgestellten Kreuzvalidierungen werden die Leistungswerte der einzelnen Durchläufe typischerweise arithmetisch gemittelt. Siehe hierzu das Beispiel des *Recalls* in den Gleichungen (5.18) und (5.19). Dies ist insofern von Bedeutung, als dass beispielsweise der F_1 -Score zwar als das harmonische Mittel aus *Precision* und *Recall* beschrieben wird (vgl. Gleichung (5.7)), gleichzeitig aber der über die Durchläufe arithmetisch gemittelte F_1 -Score nicht zwangsläufig zwischen den über die Durchläufe arithmetisch gemittelten Kennwerten *Precision* und *Recall* liegen muss.

Folgendes Beispiel verdeutlicht diesen Zusammenhang: Gegeben sei eine Gesamtstichprobe $\mathcal{D}_{\text{Gesamt}}$ mit $|\mathcal{D}_{\text{Gesamt}}| = 12$. Diese Stichprobe werde für eine 3-fache Kreuzvalidierung gemäß der Gleichungen (5.15) bis (5.17) genutzt. In jedem der Durchläufe mit Index $i = 0, 1, 2$ sei die jeweilige Teststichprobe zur Leistungsmessung eines Klassifikators dabei durch 4 Merkmalsvektoren

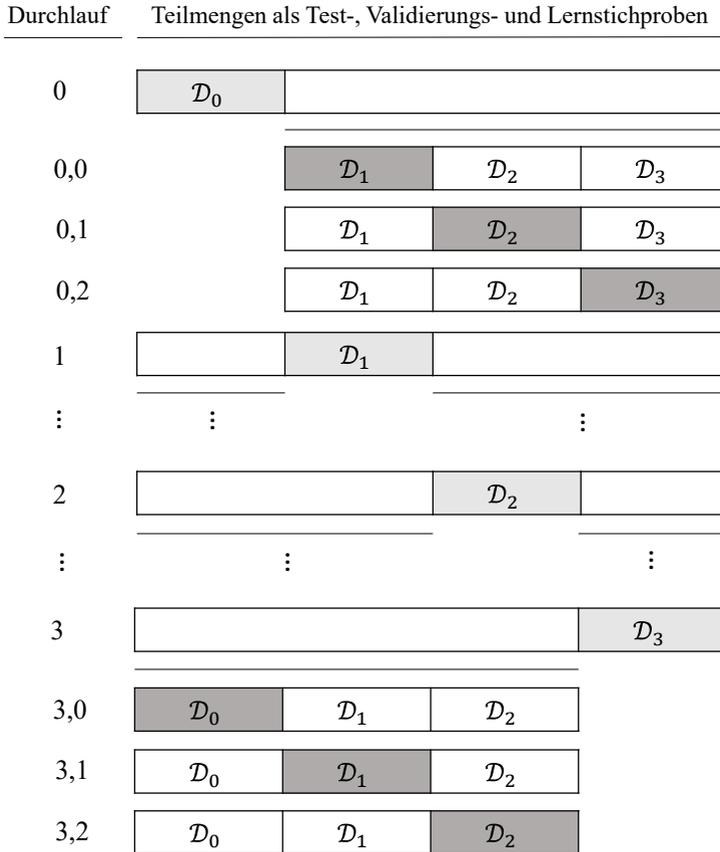


Abbildung 5.14: Die Abbildung zeigt eine 4-fache Kreuzvalidierung, in die eine 3-fache Kreuzvalidierung eingebettet wurde. In den Zeilen sind die 4 äußeren Durchläufe (0-3) sowie die insgesamt 12 inneren Durchläufe 0,0 - 3,2 vereinfacht abgebildet. Die Teststichproben sind hellgrau, die Validierungsstichproben dunkelgrau und die Lernstichproben weiß dargestellt. In jedem Durchlauf 0-3 wird eine andere Teilmenge \mathcal{D}_i mit $i = 0 \dots 3$ als Testmenge verwendet. Für die Vereinigung der jeweiligen Restmengen werden im Beispiel die Validierungs- und Lernstichproben nach dem Schema einer 3-fach Kreuzvalidierung festgelegt.

$\mathbf{x}_i^0, \dots, \mathbf{x}_i^3$ gegeben. Dabei seien die wahren und die geschätzten Klassen der 4 Merkmalsvektoren des i -ten Durchlaufs allgemein beschrieben durch:

$$O_{i,\omega} = (\omega(\mathbf{x}_i^0), \omega(\mathbf{x}_i^1), \omega(\mathbf{x}_i^2), \omega(\mathbf{x}_i^3)),$$

$$O_{i,\hat{\omega}} = (\hat{\omega}(\mathbf{x}_i^0), \hat{\omega}(\mathbf{x}_i^1), \hat{\omega}(\mathbf{x}_i^2), \hat{\omega}(\mathbf{x}_i^3)).$$

Zur Demonstration des Sachverhalts seien folgende beispielhafte wahre und geschätzte Klassen der 4 Merkmalsvektoren für die Durchläufe 0-2 gegeben, wobei der Einfachheit halber von zwei möglichen Klassen, pos. und neg., ausgegangen wird:

$$O_{0,\omega} = (\text{pos.}, \text{neg.}, \text{neg.}, \text{neg.}),$$

$$O_{0,\hat{\omega}} = (\text{pos.}, \text{pos.}, \text{pos.}, \text{pos.}),$$

$$O_{1,\omega} = (\text{pos.}, \text{pos.}, \text{pos.}, \text{pos.}),$$

$$O_{1,\hat{\omega}} = (\text{pos.}, \text{pos.}, \text{neg.}, \text{neg.}),$$

$$O_{2,\omega} = (\text{pos.}, \text{pos.}, \text{pos.}, \text{neg.}),$$

$$O_{2,\hat{\omega}} = (\text{pos.}, \text{neg.}, \text{neg.}, \text{pos.}).$$

Für jeden der Durchläufe 0-2 wird nach Schema der Gleichungen (5.5), (5.6) und (5.7) sowie (5.18), (5.19) sowohl *Recall* und *Precision* als auch F_1 -Score bestimmt.

Betrachtet man z.B. zunächst $O_{0,\omega}$ und $O_{0,\hat{\omega}}$, so lässt sich errechnen:

$$tp = 1, fp = 3, tn = 0, fn = 0.$$

Folglich ergeben sich für die *Precision* ein Wert von $\frac{1}{4}$ und für den *Recall* ein Wert von 1. Das harmonische Mittel beider Werte und damit der F_1 -Score ist gegeben durch: $\frac{2}{5}$.

Tabelle 5.2: Die Tabelle zeigt in den ersten Zeilen die Ergebnisse der Kreuzvalidierung bezüglich *Precision*, *Recall* und F_1 -Score. In den unteren Zeilen ist das arithmetische Mittel der darüber liegenden Leistungswerte der einzelnen Durchläufe angegeben.

Durchlauf	<i>Precision</i>	<i>Recall</i>	F_1 -Score
0	$\frac{1}{4}$	1	$\frac{2}{5}$
1	1	$\frac{1}{2}$	$\frac{2}{3}$
2	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{2}{5}$
Arithmet. Mittel	$\frac{1}{3} \cdot \left(\frac{1}{4} + \frac{4}{4} + \frac{2}{4}\right)$	$\frac{1}{3} \cdot \left(\frac{6}{6} + \frac{3}{6} + \frac{2}{6}\right)$	$\frac{1}{3} \cdot \left(\frac{6}{15} + \frac{10}{15} + \frac{6}{15}\right)$
	$= \frac{7}{12}$	$= \frac{11}{18}$	$= \frac{22}{45}$
	($\approx 0,583$)	($\approx 0,611$)	($\approx 0,489$)

Vollzieht man diese Rechnungen für alle Durchläufe, so ergeben sich die Leistungswerte aus Tabelle 5.2. Ersichtlich wird hier, dass der über die Durchläufe arithmetisch gemittelte F_1 -Score nicht zwischen den über die Durchläufe arithmetisch gemittelten Werten für *Precision* und *Recall* liegt, obgleich sich natürlich pro Durchlauf der F_1 -Score als harmonisches Mittel aus diesen beiden Kennwerten darstellt.

5.3 Stand der Technik

Im folgenden Abschnitt werden nennenswerte Modellierungsansätze für eine medizinische Phasendetektion vorgestellt. Dabei werden die unterschiedlichen Eigenschaften der Modelle aufgezeigt.

5.3.1 Random Forest (RF)

Die Idee hinter einem Random Forest (RF) ist es, mehrere Entscheidungsbäume in einem Wald (engl. forest) heranzuziehen und deren Ergebnisse zu kombinieren. Bezogen auf die Klassifikation ist ein RF ist also ein Modell, das seine Entscheidungen aus einzelnen Klassifikatoren (hier: Entscheidungsbäume) bezieht, die in einem Ensemble gebündelt werden (hier: Wald) – RF zählen daher zu den so genannten Ensemble-Methoden [Rok10, Zho12].

Ein RF nutzt dabei eine Menge von m Entscheidungsbäumen [Bre01]

$$\{b_i(\mathbf{x}) : i = 0, \dots, m\}. \quad (5.20)$$

Zum Training dieser Entscheidungsbäume werden zunächst verschiedene Lernstichproben nach dem Zufallsprinzip generiert. Dazu wird für jeden Baum $b_i(\mathbf{x})$ aus einer gemeinsamen, gegebenen, originalen Lernstichprobe $\mathcal{D} = \{\mathbf{x}_0, \dots, \mathbf{x}_n\}$ mit bekannten Klassenzugehörigkeiten $\omega(\mathbf{x}_j)$ für $j = 0, \dots, n$ eine eigene Lernstichprobe erzeugt. Dabei wird nach dem Prinzip Ziehen-und-Zurücklegen verfahren: Es werden $n' \leq n$ Merkmalsvektoren $\mathbf{x}_j \in \mathcal{D}$ zufällig (und ggf. mehrfach) in die jeweilige baumspezifische Lernstichprobe übernommen. Diese Vorgehensweise wird in der Statistik auch als Bootstrapping bezeichnet¹. In Kombination mit einer Aggregation (engl. aggregation) der Ergebnisse, wie im vorliegenden Falle durch die Ensemble Methode

¹ Hinweis: Der Stiefelriemen (engl. bootstrap) wird teils als Anspielung auf eine Münchhausengeschichte gesehen, in der sich der Protagonist, feststeckend in einem Sumpf, an den eigenen Stiefeln/Haaren aus der Lage befreit. Selten wird daher in diesem Zusammenhang der deutsche Begriff der Münchhausenmethode verwendet.

der RF gegeben, kann von sogenanntem Bagging gesprochen werden, das bedeutet ein Bootstrapping mit nachfolgender Aggregation. Die Idee dahinter ist, übertragen auf RF, eine große Anzahl möglichst unkorrelierter Bäume zu erzeugen (jeder Baum sieht einen anderen Teil der Lernstichprobe) und die Varianz durch Mittelwertbildung (Wald ist ein Ensemble aus gleichgewichteten Bäumen) zu senken. Eine eingängige Herleitung dieses Zusammenhangs für RF findet sich in [Fri01].

Um die Bäume des Waldes weiter zu dekorrelieren, wird bei den RF allerdings der Ansatz des Baggings noch einmal erweitert – auch hier spielt die zufällige (engl. random) Auswahl, diesmal nicht der Merkmale sondern der Merkmalswerte, eine Rolle. Das bedeutet, beim Training eines Entscheidungsbaumes wird für jeden Knoten nur eine zufällige Teilmenge der Merkmalswerte eines Merkmalsvektors für die Aufteilung in Äste des Baums berücksichtigt. Die Größe des betrachteten Merkmalsvektors \mathbf{x} ist also in jeder Teilung $d' \leq d$ (häufig wird $d' = \sqrt{d}$ verwendet). Basierend auf einem Gütekriterium (beispielsweise Entropie- oder Geni-Heterogenitätsmaß) und unter Berücksichtigung der d' Merkmalswerte, wird die Verzweigung mit höchster Güte berechnet¹.

Diese Verzweigungen teilen die Lernstichprobe sukzessive in disjunkte Teilmengen auf, die nach dem Gütekriterium bezüglich ihrer Klassenzugehörigkeit wenig heterogen sind. Ein mögliches Abbruchkriterium des Trainingsalgorithmus eines Baumes ist, neben dem Trivialfall der Erzeugung einer vollständig homogenen Teilmenge, dass eine Mindestanzahl an Stichprobenelemente in einem Knoten unterschritten wird – solche Knoten werden dann zu

¹ Hinweis: Würde für jede Teilung $d' = d$ gelten, so läge gesamt betrachtet ein Bagging vor, also ein Bootstrapping und nachfolgende Aggregation der Ergebnisse der Einzelklassifikatoren – sprich: Bagging von Bäumen. Die Idee des Baggings von Klassifikatoren wurde bereits 1996 in [Bre96] vorgestellt – insofern können RF, welche 5 Jahre später vom selben Autor in [Bre01] präsentiert wurden, als Erweiterung des Ansatzes speziell für Entscheidungsbäume gesehen werden. Diese Erweiterung betrifft die Einschränkung der Nutzung von Merkmalen – dies erscheint sinnvoll, da gut trennende Merkmale (im Hinblick auf das gewählte Gütekriterium) in vielen Entscheidungsbäumen bevorzugt in Wurzelnähe gewählt würden und somit gleichartige Bäume im Wald herangezogen würden. Die Einschränkung begünstigt also eine größere Vielfalt der Bäume.

einem Blatt des Baumes, es findet also keine weitere Verzweigung statt. Eine weitere Möglichkeit ist die Tiefe der Bäume des Waldes zu beschränken und das Training eines Baumes spätestens bei Erreichen der Maximaltiefe zu beenden.

Für die Klassifikation erhält jeder Entscheidungsbaum $b_i(\mathbf{x})$ des RF den gleichen Merkmalsvektor \mathbf{x} und bildet ihn auf eine Klasse $\omega \in \Omega$ ab. Die entsprechende a-posteriori-Wahrscheinlichkeit ist gegeben durch¹ [Bre01]:

$$\hat{P}(\omega|\mathbf{x}) = \frac{1}{m} \sum_{i=0}^{m-1} [b_i(\mathbf{x}) = \omega] . \quad (5.21)$$

Alternativ kann auch über die geschätzten Wahrscheinlichkeiten gemittelt werden [Lia02]:

$$\hat{P}(\omega|\mathbf{x}) = \frac{1}{m} \sum_{i=0}^{m-1} \hat{P}_i(\omega|\mathbf{x}) . \quad (5.22)$$

Die geschätzte Klasse $\hat{\omega}$ ist in beiden Fällen gegeben durch [Bre01]:

$$\hat{\omega}(\mathbf{x}) = \arg \max_{\omega \in \Omega} \hat{P}(\omega|\mathbf{x}) . \quad (5.23)$$

In [Sta14] wurde die Verwendung von RF-Modellen für eine medizinische Phasenerkennung am Anwendungsbeispiel einer laparoskopischen Cholezystektomie (Entfernung der Gallenblase, vgl. Kapitel 5.1.2) erstmals näher beleuchtet. Dabei wurden 7 verschiedene Phasen unterschieden. Die Lernstichprobe \mathcal{D} für die Phasenerkennung umfasste 4 durchgeführte Operationen mit bekannten Klassenzugehörigkeiten für jeden beobachteten Merkmalsvektor dieser Operationen. Die Klassenzugehörigkeiten wurden

¹ Hinweis: $[\cdot]$ stellt ein Prädikatabbildung dar, d.h. $[\cdot]$ hat einen Wert von 1, wenn der entsprechende Ausdruck \cdot als wahr ausgewertet wird. Dementsprechend wird ein Wert von 0 zugewiesen, wenn der Ausdruck als falsch ausgewertet wird.

durch medizinische Experten gesetzt. Nominale Merkmale waren die Verwendung von acht spezifischen Instrumenten (ja/nein), der Zustand des Operationslichts (an/aus) sowie der Zustand der Raumbeleuchtung (an/aus). Die Autoren erklären, dass RFs geeignet sind, da sie für Mehrklassenprobleme sehr gute Ergebnisse liefern und Phasen in a-typischer Reihenfolge vorhersagen können. Ersteres ist für realistische Arbeitsabläufe im Bereich der Medizin mit einer Anzahl an Phasen > 2 als sinnvoll zu betrachten (vgl. Kapitel 5.1 und insbesondere 5.1.2 sowie 5.1.1 für prototypische medizinische Interventionen).

Letzteres, also die Vorhersage a-typischer Reihenfolgen von Phasen, ist eine Folge der Tatsache, dass ein RF zunächst nur einen einzigen Merkmalsvektor für die Klassifikation berücksichtigt – vgl. Gleichungen (5.21), (5.22) und (5.23). Wird der Merkmalsvektor nicht künstlich erweitert, so wie in der betrachteten Arbeit von [Sta14], so bedeutet das de facto, dass ein Merkmalsvektor klassifiziert wird, ohne die Reihenfolge der Phasen zu berücksichtigen. Es wird also nicht eine Sequenz von Merkmalsvektoren modelliert und klassifiziert, sondern das Problem wird von einer Sequenz heruntergebrochen auf losgelöste, voneinander unabhängige Beobachtungen in Form von einzelnen Merkmalsvektoren, die unabhängig voneinander klassifiziert werden.

Der in [Sta14] vorgestellte RF bestehend aus 50 Bäumen mit einer maximalen Tiefe von 4 Knoten und $d' = 4$. Klassen, die durch ähnliche Merkmalswerte gekennzeichnet sind (und sich nur in der Reihenfolge des Auftretens unterscheiden) können aufgrund der Modellierung nicht unterschieden werden. In [Sch16] erhalten die Autoren vergleichbare Ergebnisse für eine vereinfachte, 7-phasige, endoprothetische Versorgung von Patienten mit einer Hüft-TEP – auch hier ist in den Konfusionsmatrizen eine hohe Verwechslung der Phasen zu beobachten die stark von der Reihenfolge abhängig sind.

Hierbei handelt es sich um ein prototypisches Verhalten eines Klassifikators, der die Abfolge der Merkmalsvektoren nicht berücksichtigt. Daher ist dieses Modell, ohne Modifikationen, hinsichtlich der Konstituenten des Problems (vgl. Kapitel 5.2) als wenig geeignet zu betrachten. Um den konzeptionellen Nachteil dennoch zu verringern, kann ein solches Modell in ein sequenzielles

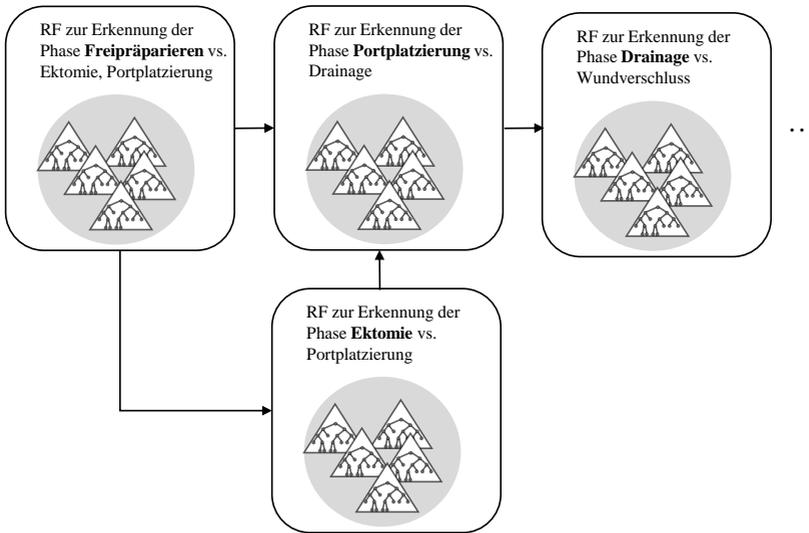


Abbildung 5.15: Die Abbildung zeigt die Nutzung eines Random Forest, der in ein einfaches sequentielles Modell eingebettet ist. Ein RF wird verwendet, um die aktuelle und mögliche nächste Phase vorherzusagen. (beispielsweise Freipräparieren mit möglichen nächsten Phasen Ektomie und Portplatzierung). Wenn eine nächste Phase vorhergesagt wird, findet ein Übergang statt. In Folge wird ein separater RF genutzt, um zwischen der neuen aktuellen Phase und möglichen nächsten Phasen zu unterscheiden. Wenn ein Übergang versehentlich gemacht wird, kann sich das Modell nicht mehr „fangen“. Daher wird ein globaler RF zur Unterscheidung zwischen allen (möglichen) Phasen verwendet, um das sequentielle Modell zurücksetzen zu können.

Modell eingebettet werden. Eine einfache Form eines derartigen Modells ist beispielsweise ein deterministischer Automat.

In [Kat16] wird dieses Konzept (vgl. Abbildung 5.15) an zwei Anwendungsbeispielen illustriert: Zum einen eine Pankreasresektion (Abschneiden der Bauchspeicheldrüse), die 12 Klassen und eine Lernstichprobe mit 11 Operationen umfasst. Zum anderen eine Adrenalektomie (Entfernung der Nebennieren) mit 9 Klassen und einer Lernstichprobe von 5 Operationen. Berücksichtigt werden die Verwendung eines spezifischen Instruments, die durchgeführte Aktion und der anatomische Aufbau.

Die Verwendung dieses sequentiellen Rahmenwerks (vgl. Abbildung 5.15) ist vergleichbar mit einem Verschieben eines Fensters über ein Modell eines Arbeitsablaufs (vgl. Abbildung 5.16), wobei das Fenster allerdings nicht nur die aktuelle Phase sondern zusätzlich mögliche nächste Phasen umfasst.

Ein fensterspezifischer Klassifikator unterscheidet dann nur noch zwischen den Phasen innerhalb des Fensters, wie auch in in [Phi16a, Phi17a] dargestellt. Das Fenster wird nach vorne (entlang der gerichteten Kante) verschoben, sobald eine nächste Phase vorhergesagt wird.

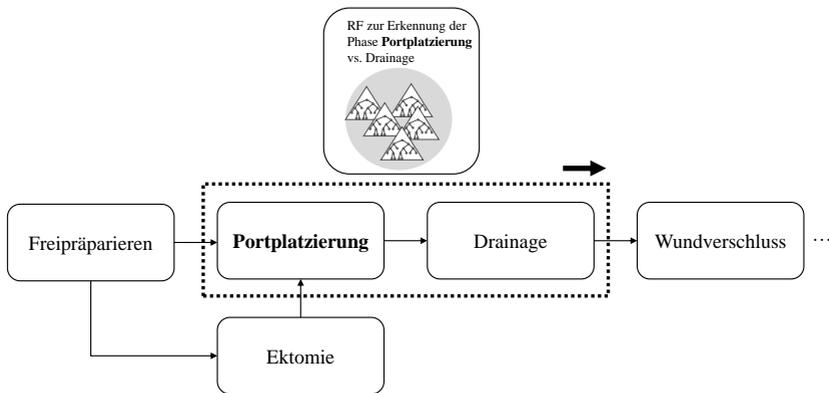


Abbildung 5.16: Das in [Kat16] vorgestellte Rahmenwerk entspricht der Nutzung eines fensterspezifischen Klassifikators. Das Fenster umfasst die aktuelle Phase und mögliche nächste Phasen. Sobald eine der möglichen nächsten Phasen vorhergesagt wird, wird das Fenster entlang des Kontrollflusses verschoben. Bei diesem Ansatz wird zwischen benachbarten Phasen diskriminiert und damit letztlich der Übergang zwischen den Phasen modelliert.

Ein entscheidender Nachteil dieses Ansatzes ist die Verwendung von deterministischen Übergängen, die es notwendig macht, Strategien zum Zurücksetzen des sequentiellen Modells zu implementieren. Im dargestellten Falle geschieht dies mittels eines sogenannten globalen RF der wie in [Sta14] über alle Phasen, unabhängig der Reihenfolge, klassifiziert. Eine Verallgemeinerung des in

[Kat16] verwendeten deterministischen Automaten sind Petri-Netze (vgl. Kapitel 3.2). Diese werden im nächsten Abschnitt genauer beleuchtet.

5.3.2 Petri-Netz

Es gibt eine ganze Reihe von Gründen, Petri-Netze [Pet62] als ein Modellierungswerkzeug für dynamische Aspekte eines Prozesses zu betrachten [van96, van98]. Im Hinblick auf das Anwendungsbeispiel können Petri-Netze aufgrund ihrer formalen Semantik, Abbildung von Nebenläufigkeiten und der Fülle von Analysetechniken von Nutzen sein.

Für eine umfängliche Einführung in die Thematik der Petri-Netze sei auf Kapitel 3.2 verwiesen. Darin wurde die Netzstruktur eines Petri-Netzes ist gegeben durch das Tupel [Rei13]

$$NST = (\mathcal{P}, \mathcal{Q}, \mathcal{F}), \quad (5.24)$$

wobei \mathcal{P} die Menge der Plätze und \mathcal{Q} die Menge der Transitionen sind

$$\begin{aligned} \mathcal{P} &= \{p_i : i = 0, \dots, |\mathcal{P}| - 1\}, \\ \mathcal{Q} &= \{q_j : j = 0, \dots, |\mathcal{Q}| - 1\}, \end{aligned}$$

so dass

$$\mathcal{P} \cap \mathcal{Q} = \emptyset.$$

Die Flussrelation \mathcal{F} gibt die Verbindung von Plätzen und Transitionen (und umgekehrt) an:

$$\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{Q}) \cup (\mathcal{Q} \times \mathcal{P}). \quad (5.25)$$

Um das dynamische Verhalten des Systems zu modellieren, werden so genannte Marken eingeführt. Die Verteilung der Marken auf der Menge der Plätze stellt den Zustand des Petri-Netzes dar (vgl. Kapitel 3.2), die initiale Markierung ist dabei gegeben durch $M_0 : \mathcal{P} \rightarrow \mathbb{N}$. Sie kann auch kompakt

in Vektor- oder Tupelform angegeben werden. Dabei repräsentieren die Einträge i des Vektors oder Tupels die Anzahl der Marken auf den Plätzen p_i . Letztere Variante wird im Folgenden zur Verdeutlichung der Zusammenhänge gewählt¹.

Nun werde ein Stochastisches Petri-Netz (SPN) definiert als ein Tupel

$$SPN = (NST, M_0, \mathcal{L}), \quad (5.26)$$

wobei $\mathcal{L} = \{\lambda_j : j = 1, \dots, |\mathcal{Q}|\}$ die Menge der Schaltraten λ_j , die den Transitionen q_j zugeordnet sind [Mol81]. Dabei sind die Schaltzeiten l exponentialverteilt, genügen somit der kumulativen Verteilungsfunktion

$$F_j(l) = 1 - e^{-\lambda_j l}. \quad (5.27)$$

Das bedeutet, dass jeder Transition q_j eine Schaltrate $\lambda_j \in (0 \dots \infty)$ zugeordnet wird, deren Kehrwert dem Erwartungswert der Schaltzeit l entspricht.

Zur Veranschaulichung zeigt Tabelle 5.3 eine graphische Repräsentation eines SPN mit

$$\begin{aligned} \mathcal{P} &= \{p_0, p_1, p_2, p_3, p_4\}, \\ \mathcal{Q} &= \{q_0, q_1, q_2, q_3, q_4\}, \\ \mathcal{F} &= \{(p_0, q_0), (q_0, p_1), (q_0, p_2), (p_1, q_1), (p_2, q_2), (q_1, p_3), \\ &\quad (q_2, q_4), (p_3, q_4), (p_3, q_3), (q_3, p_1), (p_4, q_4), (q_4, p_0)\}. \end{aligned}$$

Kreise stellen die Plätze und Rechtecke die Transitionen der Mengen \mathcal{P} und \mathcal{Q} dar. Die Flussrelation wird durch gerichtete Kanten (Pfeile) des Graphen visualisiert. Weiterhin gilt $M_0(p_1) = 1$ sowie $M_0(p_i) = 0 \quad \forall i = 2, \dots, 5$. Die

¹ Hinweis: Selten wird zur Angabe der Markierung eine Multimenge von Plätzen verwendet, hierin repräsentiert die Anzahl der Vorkommen eines Platzes die Anzahl der Marken, welche mit dem Platz assoziiert sind. Hierdurch kann die Nummerierung der Plätze entfallen, allerdings müssen der Name eines Platzes eindeutig sein

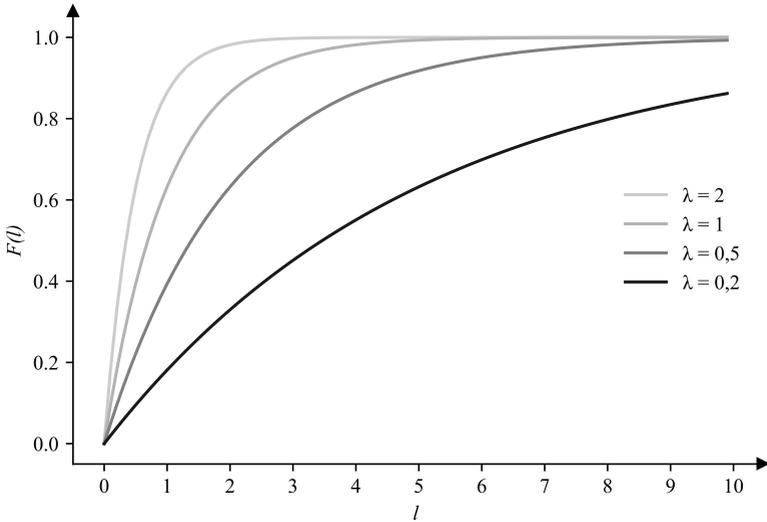


Abbildung 5.17: $F(l)$ ist die kumulative Verteilungsfunktion einer Wahrscheinlichkeitsdichtefunktion $f(l)$. Weiterhin ist l die Schaltzeit, λ^{-1} ist der Erwartungswert der Schaltzeit, $\lambda \in (0 \dots \infty)$. Die kumulative Verteilungsfunktion gibt an, wie wahrscheinlich es ist, dass bis zu einem Zeitpunkt eine Transition schon gefeuert hat.

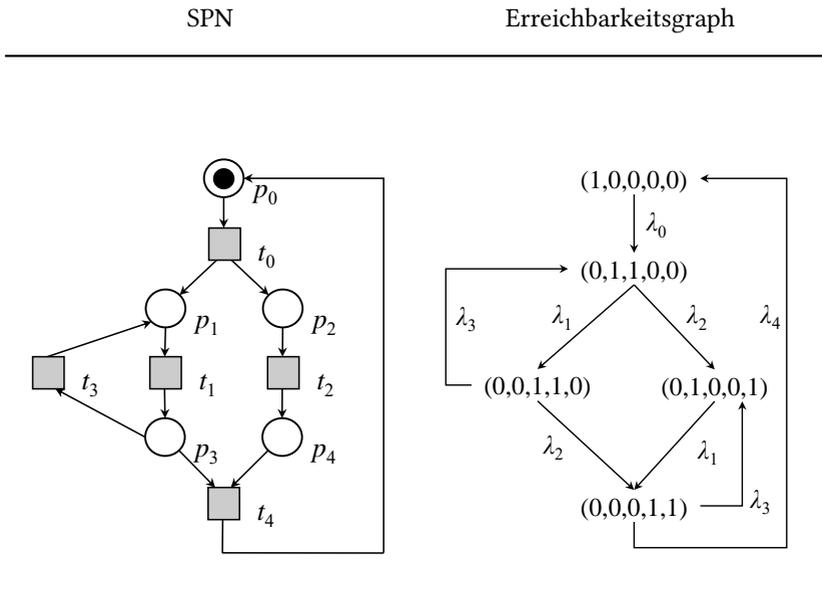
Menge der zugeordneten Schaltraten ist gegeben durch

$$\mathcal{L} = \{\lambda_0, \lambda_1, \lambda_2, \lambda_3, \lambda_4\},$$

wobei diese auf der linken Seite der Tabelle 5.3 aus Übersichtlichkeitsgründen weggelassen wurden. Die Anfangsmarkierung, also der anfängliche Zustand, ist in Tupelform gegeben durch $(1,0,0,0,0,0)$. Sprich: der Zustand $(1,0,0,0,0,0)$ des Erreichbarkeitsgraphen, vgl. rechte Seite der Tabelle 5.3, stellt den Zustand des links dargestellten SPN dar. Tabelle 5.3 veranschaulicht die Tatsache, dass ein SPN einen Markov-Prozess darstellt.

Das bedeutet, die Zustände des Erreichbarkeitsgraphen des SPN entsprechen den Zuständen eines Markov-Prozesses und die Schaltraten λ_j von Zustand

Tabelle 5.3: Die Tabelle zeigt ein SPN auf der linken Seite. Kreise stellen Plätze p_i und Rechtecke stellen Transitionen q_j dar. Der Zustand des SPN ist durch die aktuelle Verteilung der Marken (Markierung) gegeben. Im Erreichbarkeitsgraphen werden die Zustände durch Tupel dargestellt, wobei jeder Eintrag i die Anzahl der Marken an einem Platz p_i repräsentiert. Die Abbildung veranschaulicht, dass ein SPN einen Markov-Prozess darstellt, wobei unter anderem die Schaltraten λ_j von Zustand z_m nach z_n im Erreichbarkeitsgraphen der Transitionswahrscheinlichkeit a_{mn} eines (diskreten) Markov-Prozesses entsprechen. In Anlehnung an [Bau02].



z_m nach z_n des Erreichbarkeitsgraphen entspricht der Übergangswahrscheinlichkeit a_{mn} eines (diskreten) Markov-Prozesses¹. Das liegt u.a. daran, dass die Feuerraten exponentiell verteilt sind (vgl. Erinnerungslosigkeit) und damit die Markov-Eigenschaft [Bau02] erfüllt ist.

¹ Hinweis: a_{mm} wird so gesetzt, dass $a_{mn} = 1$ gilt.

Die Verwendung von Markov-Modellen mit beobachtbaren Zuständen eignet sich in Fällen, in denen die Zustände des modellierten Systems direkt zugänglich sind. beispielsweise wird in [Phi15b] ein Entscheidungsunterstützungssystem für die Diagnose von zwei komplexen Krebserkrankungen mit Hilfe von Petri-Netzen modelliert. Das involvierte Dialogsystem ermöglicht einen direkten Zugriff auf die notwendigen Merkmale. Natürlich kann auch ein solches Modell als sequentielles Modell verwendet werden, um ein Modell wie einen RF anstelle eines deterministischen Automaten einzubetten (vgl. Abschnitt 5.3.1). Im Hinblick auf die Konstituenten des Problems muss dieses Modell jedoch erweitert werden, beispielsweise durch die Einführung verborgener Zustände.

5.3.3 Hidden Markov Model (HMM)

Ein HMM [Rab86] ist ein Markov-Modell, das einen Markov-Prozess mit verborgenen, also nicht beobachteten, Zuständen repräsentiert. Das heißt, im Gegensatz zu Markov-Prozessen, bei denen der Zustand des Modells zu einem Zeitpunkt t durch dessen direkt beobachtbare Zustände bekannt ist, kann ein Beobachter in einem HMM nur auf Emissionen zugreifen, die durch die verborgenen Zustände erzeugt werden.

Um dies zu ermöglichen, ist neben der Menge der Zustände und den Zustandsübergangswahrscheinlichkeiten auch ein Beobachtungsmodell, repräsentiert durch ein Ausgabevokabular als auch Emissionswahrscheinlichkeiten, Teil eines HMM.

Formal ist ein (diskretes) HMM gegeben durch das 5-Tupel¹

$$HMM = (\mathcal{S}, \mathcal{H}, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi}). \quad (5.28)$$

¹ Hinweis: Dieses Quintupel wird in der Literatur häufig auch als $\lambda = (\mathcal{S}, \mathcal{V}, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$ bezeichnet. Um Verwechslungen zu vermeiden, sind die Tupel der verschiedenen Modelle in dieser Ausarbeitung mit sprechenden Bezeichnern versehen. Also in diesem Falle: *HMM* statt λ .

Dabei ist $\mathcal{S} = \{s_0, \dots, s_{n-1}\}$ die Menge der verborgenen Zustände des HMMs und $\mathcal{H} = \{h_0, \dots, h_{m-1}\}$ die Menge möglicher Emissionen dieser Zustände. Letzteres wird auch als Ausgabevokabular bezeichnet. Darüber hinaus existiert eine Übergangsmatrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, die die Übergangswahrscheinlichkeiten von einem aktuellen Zustand s_i zu einem nächsten Zustand s_j durch den Matrixeintrag $a_{i,j}$ bündelt. Die Matrix $\mathbf{B} \in \mathbb{R}^{n \times m}$ enthält die Emissionswahrscheinlichkeiten eines Zustandes s_i durch die entsprechenden Zeileneinträge $b_{i,j} : j = 0, \dots, m - 1$. Schließlich enthält der Vektor $\boldsymbol{\pi} \in \mathbb{R}^n$ die Anfangszustandswahrscheinlichkeiten, also die Wahrscheinlichkeit, dass ein Zustand der Anfangszustand ist.

Die Verwendung eines HMM für eine medizinische Phasenerkennung wurde in [Blu08] am Anwendungsbeispiel einer laparoskopischen Cholezystektomie (vgl. Kapitel 5.1.2) herausgearbeitet. Im Zusammenhang mit dieser Operation wurden 14 Phasen betrachtet. Die originale Lernstichprobe umfasste 12 aufgezeichnete Beobachtungssequenzen der chirurgische Eingriffe. Nominale Merkmale waren die Verwendung von 17 spezifischen Instrumenten (ja/nein). Die Autoren stellten ein HMM mit 14 Zuständen sowie ein aggregiertes HMM vor. In Letzterem wurden ähnliche Zustände in einen einzigen Zustand überführt.

In [Pad08] wird ebenso eine Cholezystektomie mit 14 Phasen betrachtet. Das Trainingsset umfasste 11 chirurgische Eingriffe. Nominale Merkmale waren die Verwendung von 18 spezifischen Instrumenten (ja/nein) einschließlich des Zustands eines optischen Geräts (eingesetzt/nicht eingesetzt).

Ein HMM kann als ein klassisches, sequenzielles Modell gesehen werden, das die Konstituenten des Problems berücksichtigt (vgl. Abschnitt 5.2). Dennoch ist die Struktur dieses Modells wenig geeignet, um Expertenwissen über Emissionswahrscheinlichkeiten einzubeziehen.

Das liegt auch daran, dass die entsprechende Wahrscheinlichkeitsverteilung nicht faktorisiert werden kann und daher exponentiell mit der Anzahl der möglichen Emissionen wächst. Um diesen Nachteil zu überwinden, kann die Verallgemeinerung eines HMM, nämlich ein DBN, verwendet werden.

5.3.4 Dynamisches Bayes-Netz (DBN)

Ein DBN ist ein probabilistisches graphisches Modell (PGM), das graphentheoretische Ansätze mit Ansätzen der Wahrscheinlichkeitstheorie verbindet. Wie in Kapitel 2.4.5 erläutert, ist ein DBN gegeben durch das Tupel

$$DBN = (B_0, B_{\rightarrow}). \quad (5.29)$$

Dabei bündelt B_0 die Struktur sowie die a-priore Wahrscheinlichkeitsverteilung über den Zufallsvariablen $\underline{X}_{0:n}$ in Zeitschritt 0. Darüber hinaus wird B_{\rightarrow} genutzt um die bedingte Wahrscheinlichkeitsverteilung über diskrete Zeitschritte t zu spezifizieren:

$$P(\underline{X}_{0:n}^t | \underline{X}_{0:n}^{t-1}) = \prod_{i=0}^n P(\underline{X}_i^t | Pa(\underline{X}_i^t)). \quad (5.30)$$

Die gemeinsame Wahrscheinlichkeitsverteilung ist gegeben durch [Mur02]:

$$P(\underline{X}_{0:n}^{0:m}) = \left(\prod_{i=0}^n P(\underline{X}_i^0 | Pa(\underline{X}_i^0)) \right) \prod_{t=1}^m \prod_{i=0}^n P(\underline{X}_i^t | Pa(\underline{X}_i^t)) \quad (5.31)$$

$$= \prod_{t=0}^m \prod_{i=0}^n P(\underline{X}_i^t | Pa(\underline{X}_i^t)). \quad (5.32)$$

Zur Verdeutlichung der Zusammenhänge kann ein DBN auch in einer sogenannten ausgerollten Form dargestellt werden (vgl. Kapitel 2.4.5, Abbildungen 2.5 bis 2.7). In diesem Zusammenhang dient B_0 als Startverteilung in Zeitschritt 0 und B_{\rightarrow} als Schablone für alle weiteren Zeitschritte – vgl. auch Gleichung (5.32).

Abbildung 5.18 zeigt links einen beliebigen Zeitschritt einer ausgerollten DBN-Struktur, welche ein HMM repräsentiert. Bezogen auf die Konstituenten des Problems (vgl. Kapitel 5.2), stellt der mit der Wurzel assoziierte Wertebereich der Zufallsvariablen $Val(\underline{X}_0^t)$ die einzelnen Phasen dar.

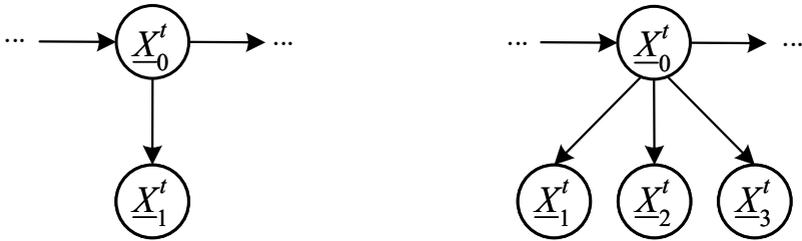


Abbildung 5.18: Die Abbildung zeigt links ein HMM, das durch eine DBN-Struktur dargestellt wird. Es gibt einen Wurzelknoten, der eine Phase repräsentiert, und einen Kindknoten, der die Emissionen einer Phase repräsentiert: $P(\underline{X}_1 | \underline{X}_0)$. Die zur Spezifizierung einer bedingten Wahrscheinlichkeitsverteilung dieses Kindknotens erforderlichen Parameter wachsen exponentiell mit der Anzahl der verwendeten Merkmale, d.h. mit der Größe des Merkmalsvektors \mathbf{x}^t . Auf der rechten Seite der Abbildung ist eine DBN-Struktur dargestellt, die die bedingte Unabhängigkeit der Merkmalswerte x_i des Merkmalsvektors \mathbf{x}^t zunutze macht, um das Wachstum der Parameter zu reduzieren: Im dargestellten Fall wächst die Anzahl der Parameter linear mit der Größe von \mathbf{x}^t .

Der Wertebereich des einzigen Kindknoten umfasst sämtliche Merkmalswertkombinationen, die beobachtet werden können - sprich: das Ausgabevokabular bzw. alle möglichen Ausprägungen des Merkmalsvektors \mathbf{x} , die einem Beobachter zur Verfügung stehen. Die Strukturen von B_0 sowie B_{\rightarrow} entsprechen denen in den Abbildungen 2.5 bis 2.7 aus Kapitel 2.4.5 - allerdings mit einem einzigen Kindknoten \underline{X} , der jeweils alle Emissionen bündelt. Die Formeln (5.29) bis (5.32) ergeben sich entsprechend.

Durch die Einführung einer bedingten Unabhängigkeit zwischen den Merkmalswerten (vgl. Abbildung 5.18, rechte Seite) kann ein naives DBN konstruiert werden. Die entsprechende Wahrscheinlichkeitsverteilung der Kinder ist durch $\prod_{i=1}^3 P(\underline{X}_i | \underline{X}_0)$ gegeben. Jede dieser Kinder repräsentiert dabei ein Merkmal x_i des Merkmalsvektors \mathbf{x} . Es ist leicht zu erkennen, dass im ersten Fall die Anzahl der Parameter exponentiell wächst, während dies im zweiten Fall linear stattfindet.

Seien beispielsweise 7 verschiedene Phasen gegeben - mithin gilt also: $|\text{Val}(\underline{X}_0^t)| = 7$. Weiterhin wird zu jedem Zeitschritt t ein Merkmalsvektor \mathbf{x}

mit Dimension $d = 3$ beobachtet, d.h. $\mathbf{x}^t = (x_0^t, x_1^t, x_2^t)^T$. Jedes der drei Merkmale habe zudem 4 verschiedenen Merkmalswerte, welche angenommen werden können. Dann ist die bedingte Wahrscheinlichkeit für das Vorliegen einer konkreten Ausprägung eines Merkmalsvektors gegeben die Phase, also $P(\underline{X}_1 | \underline{X}_0)$, gegeben durch: $(4 \cdot 4 \cdot 4) \cdot 7 = 448$ Parameter. Hiervon müssen 441 angegeben werden, weil sich die Wahrscheinlichkeitswerte zu 1 summieren. Auf der anderen Seite ist $\prod_{i=1}^3 P(\underline{X}_i | \underline{X}_0)$ gegeben durch $(4 + 4 + 4) \cdot 7 = 84$ Parameter, von denen 63 spezifiziert werden müssen. Weitere Einzelheiten finden sich in [Phi17d].

Diese Reduzierung der Parameter kann nützlich sein, wenn nur eine kleine Menge an Trainingsdaten zur Verfügung steht [Kol09]. Darüber hinaus ermöglicht die spezielle Struktur des Netzes eine sinnvolle expertenbasierte Parametrisierung und damit die Einbeziehung des Expertenwissens in die Modelle – was auch beim Umgang mit kleinen Mengen von Trainingsdaten und im Zusammenhang mit Übersetzungs- und/oder Fusionsmechanismen von Ablaufmodellen nützlich sein kann [Phi15c, Phi16b, Phi17d].

5.3.5 Zusammenfassung

Die wichtigsten Eigenschaften der vorgestellten Modellierungsansätze der vorangegangenen Kapitel 5.3.1 bis 5.3.4 sind in Tabelle 5.4 kompakt visualisiert.

Dabei zeigen die RF ein prototypisches Verhalten eines Klassifikators, der die Sequenz der Merkmalsvektoren \mathbf{x}^t während der Klassifikation nicht berücksichtigt. Auch wenn dieser konzeptuelle Nachteil abgefedert werden kann – beispielsweise indem der Klassifikator in ein sequentielles Modell eingebettet wird oder Merkmalsvektoren gestapelt werden – sind die RFs der Kategorie der nicht-sequentuellen Modelle zuzuordnen.

Weiterhin erlauben RF nicht die Modellierung verborgener Zustände, was notwendig sein kann um die Tatsache zu repräsentieren, dass ein Beobachter keine direkten Zugriff auf den Zustand des modellierten Systems hat.

Darüber hinaus ist ein RF ein diskriminatives Modell, welches eine Klassifikation zur Laufzeit (engl. online) ermöglicht. Letztere ist notwendig um eine Phasenerkennung unter Kenntnis der bisher beobachteten Merkmalsvektoren aber ohne Kenntnis der noch zukünftig kommenden zu ermöglichen (vgl. Kapitel 5.2).

Ein SPN repräsentiert einen Markov Prozess (vgl. Section 5.3.2). Das Modell ist geeignet um sequentielle Abhängigkeiten für Systeme zu modellieren, in denen der interne Zustand für einen Beobachter direkt sichtbar ist (beispielsweise in Dialogsystemen). Bei einem SPN handelt es sich um ein generatives Modell, welches eine Klassifikation zur Laufzeit erlaubt – ein datengetriebenes Einlernen ist beispielsweise über genetische Algorithmen möglich. Schließlich ist eine expertenbasierte Parametrisierung möglich – in der Praxis kann allerdings die Größe des Modells schnell zu einem limitierenden Faktor werden. In der praktischen Verwendung können deswegen Erweiterungen der Petri-Netze, beispielsweise in Form von gefärbten Petri-Netzes, notwendig werden.

Ein HMM kann als ein typisches, gut erforschtes sequentielles Modell gesehen werden. Ein HMM erlaubt die direkte Modellierung von verborgenen Zuständen, welche bezüglich der Konstituenten des Problems als notwendig betrachtet werden muss (vgl. Kapitel 5.2). Es handelt sich um einen generativen Ansatz der zur Klassifikation zur Laufzeit geeignet ist. Allerdings ist die Struktur der Modells nicht geeignet um Expertenwissen über praxisrelevante Emissionswahrscheinlichkeiten einzubringen. Das liegt unter anderem daran, dass die entsprechende Wahrscheinlichkeitsverteilung im HMM nicht faktorisiert werden kann und deswegen exponentiell mit der Anzahl der betrachteten Emissionen steigt.

Um diesen Nachteil zu überwinden können DBNs eingesetzt werden, die eine Verallgemeinerung der HMM darstellen. DBNs erweitern bzw. verallgemeinern das Modell um die Möglichkeit den Zustandsraum in faktorisierter Form dazustellen und nicht nur in Form einer einzelnen Zufallsvariablen (vgl. Kapitel 5.3.4, Abbildung 5.18). Dies erlaubt es, die Parameter zu reduzieren, was die Interpretierbarkeit und Modularität verbessert.

Tabelle 5.4: Die Tabelle zeigt verschiedene Modellierungsansätze die in diese Arbeit besprochen werden. Die Symbole ■ / □ werden genutzt um darzustellen dass die entsprechende Eigenschaft vorhanden / nicht vorhanden ist.

	RF	SPN	HMM	DBN
Sequentielles Modell	□	■	■	■
Verborgene Zustände	□	□	■	■
Generativer Ansatz	□	■	■	■
Klassifikation zur Laufzeit	■	■	■	■
Datengetriebenes Training	■	■	■	■
Expertenbasierte Parametrisierung	□	■	□	■

Dies erlaubt die Etablierung eines praxistauglichen Weges zur Einbringung von Expertenwissen in ein DBN. Weiterhin können mit DBNs, im Gegensatz zu Kalman Filter Modellen, beliebige Wahrscheinlichkeitsverteilungen (und nicht nur unimodale, lineare Gaußverteilungen) festgelegt werden. Ein DBN ist in Summe eine vielversprechendes Modell, da es eine Abwägung zwischen Ausdrucksstärke und Komplexität erlaubt und bewährte probabilistische Modelle wie HMM durch seine Verallgemeinerung mit umfasst.

In jüngster Zeit rücken Künstliche Neuronale Netze (KNN) auch für den Bereich der medizinischen Phasenerkennung in den Fokus des Interesses. Beispielsweise wird in [Li16] ein Faltendes Neuronales Netz (engl. Convolutional Neural Network) (CNN) verwendet, um drei Phasen am Anwendungsbeispiel einer Wiederbelebensmaßnahme mit 3 Phasen zu erkennen. Zum Einlernen des Klassifikators wurden 20 Arbeitsabläufe in Form von Tiefenbildern und Tonaufnahmen genutzt. Das Modell wurde dann über Schwellwerte in ein einfache sequentielles Modell eingebettet – dies entspricht der Vorgehensweise aus Kapitel 5.3.1.

In [Twi17] wird ebenso ein CNN betrachtet – Anwendungsbeispiel ist hier eine Cholezystektomie mit 7 Phasen. Für das Training wurden 80 Eingriffe betrachtet. Die Autoren nutzen das CNN zur Merkmalsextraktion aus Videoendoskopaufnahmen. Mit den gewonnen Merkmalen wurde ein HMM eingelernt, welches eine Genauigkeit bei der Phasenerkennung von 82% erreichte – vermehrte Fehlklassifikationen traten an einer Verzweigungen des betrachteten Arbeitsablaufes auf.

Schließlich beleuchten die Autoren in [Li17] das Anwendungsbeispiel einer Wiederbelebensmaßnahme – hier wird ein CNN als Merkmalsextraktor für Tiefenbilder, Tonaufnahmen und Posen von Instrumenten genutzt. Nachgeschaltet kommt zur sequentiellen Modellierung ein Rekurrentes Neuronales Netz (RNN) mit spezieller Speicherarchitektur – einem sogenannten langem Kurzzeitgedächtnis (engl. Long Short-Term Memory) (LSTM) zum Einsatz. Zum Training werden 42 verschiedene Aufnahmen von Wiederbelebensmaßnahmen verwendet. Der Fokus der Arbeiten liegt dabei aber nicht auf einer tatsächlichen Phasenerkennung (vgl. Kapitel 5.2), sondern in der Erkennung 35 verschiedener Einzelaktivitäten.

Im Hinblick auf die Konstituenten des Problems (vgl. Kapitel 5.2 können KNN als Merkmalsextraktoren verwendet werden (z.B. bei Tiefenbildern). Allerdings liegen für die medizinischen Anwendungsfälle vergleichsweise wenig Trainingsdaten vor – was einen sinnvollen Einsatz der KNN erschwert. Auch wenn die betrachteten medizinischen Anwendungsfälle eher datenarm sind, so existiert jedoch in deren Umfeld ein reichhaltiges Wissen – sie sind also wissensreich. Ein Grund hierfür ist die Existenz einer Vielzahl an medizinischen Experten, die in diese Anwendungsfälle involviert sind.

5.4 Transformation eines FHIR-Plans in ein DBN zum Abgleich chirurgischer Interventionen

Im folgenden Unterkapitel 5.4.1 werden zunächst FHIR-Pläne als Ausgangsmodell erläutert. Danach wird in Unterkapitel 5.4.2 auf DBNs als Zielmodell eingegangen. Schließlich wird in Unterkapitel 5.4.3 vorgestellt, wie ein FHIR-Plan in ein DBN zur chirurgischen Phasendetektion transformiert werden kann. Die Vorgehensweise bettet sich in das in Kapitel 4.1 vorgestellte Gesamtkonzept ein, vgl. Abbildung 4.1 auf Seite 82.

5.4.1 FHIR-Plan als Ausgangsmodell

Ausgangspunkt der Transformation sei ein FHIR Plan einer chirurgischen Intervention, welcher durch einen Graphen $H = (\mathcal{K}, \mathcal{M})$ mit Knotenmenge \mathcal{K} und Kantenmenge \mathcal{M} gegeben sei. Die Knotenmenge \mathcal{K} lasse sich partitionieren in die folgenden disjunkten Knotenmengen:

- \mathcal{J} : Menge der elementaren Aktionen,
- \mathcal{J}_τ : Menge der stillen Aktionen,
- \mathcal{G}_\rightarrow : Menge der gruppierenden Aktionen für Sequenzen,
- \mathcal{G}_\times : Menge der gruppierenden Aktionen für Entscheidungen,

- $\mathcal{G}_{\mathcal{O}}$: Menge der gruppierenden Aktionen für Iterationen,
- \mathcal{O}_{Imp} : Menge der Dateninputs,
- \mathcal{O}_{Out} : Menge der Datenoutputs.

Zudem beschreibe $\mathcal{G} = \mathcal{G}_{\rightarrow} \cup \mathcal{G}_{\times} \cup \mathcal{G}_{\mathcal{O}}$ die Menge aller gruppierenden Aktionen.

Weiterhin gebe es spezielle Aktionen $\mathcal{J}_{\text{Start}}$ und $\mathcal{J}_{\text{Ende}}$ aus der Menge der elementaren Aktionen \mathcal{J} , welche die eindeutigen Start- und Endaktionen des FHIR-Plans markieren.

Die Menge der gerichteten Kanten des FHIR-Plans sei gegeben durch:

$$\mathcal{M} = \{(e_i, e_j) \mid (e_i, e_j) \in ((\mathcal{K} \setminus (\mathcal{O}_{\text{Imp}} \cup \mathcal{O}_{\text{Out}})) \times \mathcal{K}, e_i \neq e_j)\}.$$

Es existieren also keine Kanten ausgehend von Dateninputs oder Datenoutputs. Darüber hinaus gibt es keine Kanten von einem Knoten zurück in sich selbst – der Graph ist also frei von Schlingen (schlingenfrei).

Die Kanten \mathcal{M} können weiter partitioniert werden in:

- \mathcal{M}_{Sub} : Menge der Kanten, die gruppierende Aktionen mit ihren Subaktionen verbinden,
- \mathcal{M}_{Vor} : Menge der Kanten, die repräsentieren, dass eine Aktion vor dem Ende einer anderen Aktion ausgeführt werden muss (before-end),
- $\mathcal{M}_{\text{Nach}}$: Menge der Kanten, die repräsentieren, dass eine Aktion nach dem Ende einer anderen Aktion ausgeführt werden muss (after-end),
- \mathcal{M}_{Imp} : Menge der Kanten, die Aktionen mit ihren zugehörigen Dateninputs verbinden,
- \mathcal{M}_{Out} : Menge der Kanten, die Aktionen mit ihren zugehörigen Datenoutputs verbinden.

Zudem sei als Hilfe für die spätere Transformation folgende Knotenmenge definiert:

- $Desc_{H_{\text{Sub}}}(\mathbf{g})$: Nachfahren des Knoten $\mathbf{g} \in \mathcal{G}$ bezüglich des Teilgraphen H_{Sub} mit Knotenmenge \mathcal{K} und Kantenmenge \mathcal{M}_{Sub} . Diese Menge gibt alle direkten und indirekten Subaktionen einer gruppierenden Aktion \mathbf{g} an.
- $Pa_{H_{\text{Sub}}}(k)$: Elternknoten eines Knotens $k \in \mathcal{K}$ bezüglich des Teilgraphen H_{Sub} mit Knotenmenge \mathcal{K} und Kantenmenge \mathcal{M}_{Sub} .

5.4.2 DBN als Zielmodell

Ziel der Transformation sei ein DBN (vgl. Kapitel 2.4.5) welches gegeben ist durch das Tupel

$$DBN = (B_0, B_{\rightarrow}).$$

Hierbei sind B_0 und B_{\rightarrow} ihrerseits Bayes-Netze. $B_0 = (G_0, P_0)$ gibt die Struktur, sowie die a-priori Wahrscheinlichkeitsverteilung über den Zufallsvariablen in Zeitschritt 0 an. $B_{\rightarrow} = (G_{\rightarrow}, P_{\rightarrow})$ spezifiziert Struktur als auch die bedingte Wahrscheinlichkeitsverteilung über zwei diskrete Zeitschritte t und $t - 1$ hinweg.

Die Knotenmenge des DBN lasse sich aufteilen in:

- \mathcal{V}_{B_0} : Knoten des Graphen G_0 ,
- $\mathcal{V}_{B_{\rightarrow}}$: Knoten des Graphen G_{\rightarrow} .

Weiterhin gebe es in den Knotenmengen jeweils ausgezeichnete Wurzelknoten $v_{\text{Wurz}} \in \mathcal{V}_{B_0}$ und¹ $v_{\text{Wurz}}^* \in \mathcal{V}_{B_{\rightarrow}}$.

Die Kantenmengen des DBN seien gegeben durch:

- \mathcal{E}_{B_0} : Kanten des Graphen G_0 ,
- $\mathcal{E}_{B_{\rightarrow}}$: Kanten des Graphen G_{\rightarrow} .

¹ Hinweis: Das Symbol * wird hier verwendet um Mehrdeutigkeiten zu vermeiden und um einen Knoten zu markieren, der zum Zeitschritt t des Graphen B_{\rightarrow} gehört (vgl. Beispiel eines 2TBN in Abbildung 2.6 auf Seite 32).

Mithilfe der Wurzelknoten können die Kantenmengen wie folgt gebildet werden:

$$\begin{aligned}\mathcal{E}_{B_0} &= (\{u_{\text{Wurzel}}\} \times (\mathcal{V}_{B_0} \setminus \{u_{\text{Wurzel}}\})), \\ \mathcal{E}_{\rightarrow} &= (\{u_{\text{Wurzel}}^*\} \times (\mathcal{V}_{B_{\rightarrow}} \setminus \{u_{\text{Wurzel}}^*\})) \cup (\{u_{\text{Wurzel}}\} \times \{u_{\text{Wurzel}}^*\}).\end{aligned}$$

Sprich: die Wurzelknoten sind also jeweils mit ihren Kindknoten verbunden, weiterhin sind die Wurzelknoten zweier aufeinanderfolgender Zeitschritte durch eine Kante verbunden (vgl. Abbildung 2.7 auf Seite 33).

Neben den Knoten- und Kantenmengen, welche die Struktur des DBN festlegen, sind die bedingten Wahrscheinlichkeitsverteilungen von Bedeutung (vgl. Kapitel 2.4.5). Diese lassen sich für ein DBN mit k Kindern wie folgt aufteilen (vgl. Abbildung 2.7 auf Seite 33 und [Phi17a]):

$$P(\underline{X}_0^0 = i) = \boldsymbol{\pi}(i) \quad (5.33)$$

$$P(\underline{X}_0^t = j \mid \underline{X}_0^{t-1} = i) = \mathbf{A}(i, j) \quad (5.34)$$

$$P(\underline{X}_1^t = j \mid \underline{X}_0^t = i) = \mathbf{B}^1(i, j) \quad (5.35)$$

⋮

$$P(\underline{X}_m^t = j \mid \underline{X}_0^t = i) = \mathbf{B}^k(i, j) \quad (5.36)$$

Dabei zeigt (5.33) die initiale Wahrscheinlichkeitsverteilung, die mit dem Wurzelknoten \underline{X}_0 zum Zeitschritt $t = 0$ assoziiert ist, sprich: \underline{X}_0^0 . Folglich bündelt der Vektor $\boldsymbol{\pi}(i)$ die a-priori Wahrscheinlichkeiten der jeweiligen Werte von \underline{X}_0^0 . Hinweis: Das Ergebnis $X = i$ ist eine Kurzschreibweise für das Ereignis $\omega \in \Omega : f_{\underline{X}}(\omega) = i$. Dabei ist Ω die Ergebnismenge und $f_{\underline{X}}(\omega)$ bildet ein Ergebnis ω auf mögliche Werte der Zufallsvariablen \underline{X} ab. Dieser Wertebereich von \underline{X} ist durch $\text{Val}(\underline{X})$ gegeben. In Verbindung mit $P(\cdot)$ ergibt sich so $P(\underline{X} = i)$, vgl. Kapitel 2.4.2. Im gegebenen Falle beinhaltet $\text{Val}(\underline{X})$ die möglichen Phasen der chirurgischen Intervention, d.h. die möglichen Zustände.

In (5.34) ist die Wahrscheinlichkeit angegeben, um von einem Zustand i eines Zeitschrittes in einen Zustand j des darauffolgenden Zeitschrittes überzugehen (Übergangswahrscheinlichkeit). Das bedeutet, dass Matrix $\mathbf{A}(i,j)$ letztlich eine Adjazenzmatrix widerspiegelt, welche für Einträge > 0 mit Übergangswahrscheinlichkeiten erweitert wurde, vgl. hierzu auch die Kapitel 5.3.2 und 5.3.3 sowie 5.3.4.

(5.35) und (5.36) zeigen die Emissionswahrscheinlichkeiten eines Zustandes i . Im Unterschied zu einem HMM werden diese Wahrscheinlichkeiten aber nicht in einer einzigen Matrix $\mathbf{B}(i,j)$ gebündelt, vgl. Kapitel 5.3.3 und [Phi17a] – vielmehr werden die Wahrscheinlichkeitsverteilungen in faktorisierter Form, hinterlegt. Sprich: Jede Emission oder jedes Merkmal hat eine eigene Verteilung gegeben den jeweiligen Zustand des Elternknotens.

5.4.3 Transformation eines FHIR-Plans in ein DBN

Zur Transformation eines FHIR-Plans $H = (\mathcal{K}, \mathcal{M})$ in ein DBN werde zunächst eine surjektive Funktion definiert, welche die Menge $(\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}})$ der Dateninput- und Datenoutputknoten des FHIR-Plans auf eine Teilmenge der Knoten des DBN abbildet:

$$h_{\text{Map}} : (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}}) \rightarrow (\mathcal{V}_{B_0} \setminus \{v_{\text{Wurz}}\}). \quad (5.37)$$

Bei h_{Map} handele es sich um eine surjektive Abbildung, welche Datenin- und outputknoten o mit den selben Namen, d.h. gleichen Inhalts des Attributs *text*, auf den gleichen Knoten im DBN abbildet:

$$h_{\text{Map}}(o_1) = h_{\text{Map}}(o_2) \Leftrightarrow o_1.\text{text} = o_2.\text{text} \quad (5.38)$$

In der Knotenmenge \mathcal{V}_{B_0} des DBN gibt es also für jeden eindeutigen Namen der Dateninput- und -outputknoten des FHIR-Plans einen entsprechenden Knoten.

Der Wurzelknoten v_{Wurz} ist der einzige Knoten aus \mathcal{V}_{B_0} , der von h_{Name} nicht getroffen wird und kein Urbild innerhalb des FHIR-Plans besitzt. Dieser Knoten wird separat erzeugt.

Weiterhin gebe es eine surjektive Funktion h_{Map}^* mit

$$h_{\text{Map}}^* : (\mathcal{O}_{\text{Inp}} \cup \mathcal{O}_{\text{Out}}) \rightarrow (\mathcal{V}_{B_{\rightarrow}} \setminus \{v_{\text{Wurz}}^*\}), \quad (5.39)$$

welche jeden Dateninput- und Datenoutputknoten des FHIR-Plans auf Partnerknoten im Graphen G_{\rightarrow} abbildet. Auch hier werde der Wurzelknoten v_{Wurz}^* später separat erzeugt und es gelte:

$$h_{\text{Map}}^*(o_1) = h_{\text{Map}}^*(o_2) \Leftrightarrow o_1.\text{text} = o_2.\text{text} \quad (5.40)$$

Mithilfe der Funktionen h_{Map} und h_{Map}^* , sowie weiteren deterministischen Transformationsregeln können schließlich die Graphen G_0 und G_{\rightarrow} des DBN wie folgt generiert werden:

$$\mathcal{V}_{B_0} = \{h_{\text{Map}}(o) \mid o \in \mathcal{O}_{\text{Code}}\} \cup \{v_{\text{Wurz}}\}, \quad (5.41)$$

$$\mathcal{V}_{B_{\rightarrow}} = \{h_{\text{Map}}^*(o) \mid o \in \mathcal{O}_{\text{Code}}\} \cup \{v_{\text{Wurz}}^*\} \cup \mathcal{V}_{B_0}, \quad (5.42)$$

$$\mathcal{E}_{B_0} = \{(v_{\text{Wurz}}, h_{\text{Map}}(o)) \mid o \in \mathcal{O}_{\text{Code}}\}, \quad (5.43)$$

$$\mathcal{E}_{B_{\rightarrow}} = \{(v_{\text{Wurz}}^*, h_{\text{Map}}^*(o)) \mid o \in \mathcal{O}_{\text{Code}}\} \cup \{(v_{\text{Wurz}}, v_{\text{Wurz}}^*)\}. \quad (5.44)$$

Zunächst wird in (5.41) die Knotenmenge des Graphen G_0 des DBN erzeugt. Hierzu wird jedem eindeutigen Code eines Dateninputs und Datenoutputs des FHIR-Plans.

Sprich: jedem Element aus der Menge $\mathcal{O}_{\text{Code}}$ – ein entsprechender Knoten aus der Knotenmenge \mathcal{V}_{B_0} des DBN zugeordnet. Weiterhin ist der ausgezeichnete Wurzelknoten v_{Wurz} , der kein Urbild im FHIR-Plans besitzt, Teil der Knotenmenge von G_0 .

Anschließend werden in (5.42) mithilfe der Funktion h_{Map}^* in gleicher Weise die Knoten des Graphen G_{\rightarrow} des DBN erzeugt. Zusätzlich sind v_{Wurz}^* und \mathcal{V}_{B_0} Teil der Knotenmenge $\mathcal{V}_{B_{\rightarrow}}$.

Die Kantenmengen werden in (5.43) und (5.44) erzeugt. Hierbei wird in (5.43) zunächst die Kantenmenge \mathcal{E}_{B_0} erzeugt, indem der Wurzelknoten mit jedem anderen Knoten der Knotenmenge \mathcal{V}_{B_0} über eine Kante verbunden wird.

Ebenso wird in (5.44) für die Kantenmenge \mathcal{E}_{B_-} verfahren: Auch hier wird eine gerichtete Kante vom Wurzelknoten zu jedem der anderen Knoten der Menge \mathcal{V}_{B_-} gezogen.

Weiterhin müssen noch Kanten über die Zeitschritte hinweg erzeugt werden. Hierfür werden die Wurzelknoten benachbarter Zeitschritte verbunden.

Nachdem die Struktur des DBN erzeugt ist, muss nun noch dessen Parametrierung erfolgen. Zunächst werden dabei diejenigen Werte der (diskreten) Zufallsvariablen festgelegt, die durch die Knoten v_{Wurz}^* und v_{Wurz} repräsentiert werden.

Seien \underline{X}_0^t die Zufallsvariablen, die für einen Zeitschritt t mit v_{Wurz}^* assoziiert sind und \underline{X}_0^{t-1} die Zufallsvariablen, die zum Zeitschritt $t - 1$ durch v_{Wurz} repräsentiert werden. Die Werte, die diese Zufallsvariablen annehmen können, entsprechen natürlichen Zahlen, wobei jedem möglichen Prozessschritt, der durch eine Aktion repräsentiert wird, eine eindeutige natürliche Zahl zugewiesen wird. Dies geschieht durch eine bijektive Funktion:

$$h_N : \mathcal{J} \rightarrow \{0, \dots, n - 1\} \quad \text{mit } n = |\mathcal{J}|. \quad (5.45)$$

Ohne Beschränkung der Allgemeinheit wird angenommen, dass die von h_N zugewiesenen Zahlen einer lexikographischen Ordnung entsprechen. Man erhält:

$$\begin{aligned} \text{Val}(\underline{X}_0^t) &= \{h_N(u) \mid u \in \mathcal{J}\}, \\ \text{Val}(\underline{X}_0^{t-1}) &= \{h_N(u) \mid u \in \mathcal{J}\}, \end{aligned}$$

wobei $\text{Val}(\cdot)$ die Wertemenge angibt, die die jeweilige Zufallsvariable annehmen kann.

Für die anderen Zufallsvariablen $\underline{X}_{1:m}$, die den Kindknoten von v_{Wurz} und v_{Wurz}^* zugeordnet sind, wird eine Bernoulli-Verteilung verwendet, d.h.:

$$\text{Val}(\underline{X}_i^t) = \{0, 1\},$$

wobei $i = 1, \dots, m$ mit $m = |\mathcal{V}_{B_{\rightarrow}} \setminus (\{u_{\text{Wurz}}^*\} \cup \mathcal{V}_{B_0})|$. Außerdem werden die Startverteilung und die CPD der Beobachtungen und der Zustandsübergänge festgelegt. Dabei ist der Vektor der Anfangszustandswahrscheinlichkeiten π gegeben durch:

$$\pi(h_N(u)) = \begin{cases} 1, & \text{falls } u \in \mathcal{J}_{\text{Start}}, \\ 0, & \text{sonst.} \end{cases} \quad (5.46)$$

Wobei durch h_N wie in Gleichung (5.45) definiert, jeder Aktion $j \in \mathcal{J}$ eine eindeutige natürliche Zahl zugeordnet wird.

Da es nur einen dedizierten Startzustand gibt, ist die Wahrscheinlichkeit für dessen Auftreten als Anfangszustand 1. Allen anderen Zuständen wird eine Wahrscheinlichkeit von 0 zugeordnet (5.46).

Die Wahrscheinlichkeitsverteilung der Zustandsübergänge ist durch (5.47) gegeben. Ihre Darstellung entspricht einer Adjazenzmatrix mit den Übergangswahrscheinlichkeiten als Kantengewichten. Dabei sind die Einträge $\mathbf{A}(i,j)$ der Matrix \mathbf{A} mit $i = h_N(u)$ und $j = h_N(v)$ gegeben.

Hierbei gibt der Wert $u.du$ die Dauer an, die der Aktion $u \in \mathcal{J}$ über das Attribut *action.timing.timingDuration* in FHIR zugeordnet ist. Der Wert $v.trans$ gibt die Wahrscheinlichkeit eines Übergangs in die Aktion v an und ist in FHIR über das Attribut *action.transition* gegeben.

In (5.47) wird in den ersten beiden Fällen die Übergangswahrscheinlichkeit von einer Aktion zurück zu sich selbst gesetzt – sprich die Wahrscheinlichkeit in dieser Aktion zu bleiben. Dabei muss unterschieden werden, ob es sich um eine Endaktion handelt oder nicht.

Für eine Endaktion ist kein weiterer Übergang in eine Folgeaktion möglich. Die Übergangswahrscheinlichkeit aus einer Endaktion heraus ist somit 0. Umgekehrt ist die Wahrscheinlichkeit in der Endaktion zu verbleiben 1. Für eine Aktion mit Folgeaktion wird die Wahrscheinlichkeit in dieser Aktion zu verbleiben mithilfe deren Dauer $u.du$ berechnet.

In den folgenden Fällen wird schließlich die Übergangswahrscheinlichkeit von einer Aktion u in eine andere Aktion v gesetzt. Dabei kommt es darauf an, wie die beiden Aktionen u und v miteinander verschaltet sind.

Zunächst werden sequentiell miteinander verschaltete elementare Aktionen betrachtet – vgl. dritter Fall in (5.47). Solcherlei Aktionen sind im FHIR-Plan durch Kanten aus der Menge $\mathcal{M}_{\text{Nach}}$ direkt verbunden. Die Wahrscheinlichkeit in die Folgeaktion zu wechseln ist gerade durch die Gegenwahrscheinlichkeit des zweiten Falles gegeben.

$$\mathbf{A}(i, j) = \begin{cases} 1, & \text{falls } u = v, u \in \mathcal{J}_{\text{Ende}} \\ \frac{u \cdot du - 1}{u \cdot du}, & \text{falls } u = v, u \in (\mathcal{J} \setminus \mathcal{J}_{\text{Ende}}) \\ 1 - \frac{u \cdot du - 1}{u \cdot du}, & \text{falls } u \neq v, \{u, v\} \subseteq \mathcal{J}, (v, u) \in \mathcal{M}_{\text{Nach}} \\ \\ \left(1 - \frac{u \cdot du - 1}{u \cdot du}\right) \cdot v \cdot \text{trans}, & \text{falls } u \neq v, \{u, v\} \subseteq \mathcal{J}, \\ & \exists (g, u) \in \mathcal{M}_{\text{Nach}}, g \in \mathcal{G}, v \in \text{Desc}_{H_{\text{Sub}}}(g), \\ & \nexists (k_1, k_2) \in \mathcal{M}_{\text{Nach}}, k_1 \in \mathcal{W}_{H_{\text{Sub}}}^{g \rightarrow v}, k_2 \in \mathcal{K} \\ \\ 1 - \frac{u \cdot du - 1}{u \cdot du}, & \text{falls } u \neq v, \{u, v\} \subseteq \mathcal{J}, \\ & \exists (v, g) \in \mathcal{M}_{\text{Nach}}, g \in \mathcal{G}, u \in \text{Desc}_{H_{\text{Sub}}}(g), \\ & \nexists (k_1, k_2) \in \mathcal{M}_{\text{Nach}}, k_2 \in \mathcal{W}_{H_{\text{Sub}}}^{g \rightarrow u}, k_1 \in \mathcal{K} \\ \\ 1 - \frac{u \cdot du - 1}{u \cdot du}, & \text{falls } u \neq v, \{u, v\} \subseteq \mathcal{J}, \{(\tau, u), (v, \tau)\} \subseteq \mathcal{M}_{\text{Nach}}, \tau \in \mathcal{J}_{\tau} \\ \\ \left(1 - \frac{u \cdot du - 1}{u \cdot du}\right) \cdot \tau \cdot \text{trans}, & \text{falls } u \neq v, \{u, v\} \subseteq \mathcal{J}, \\ & \exists (g_1, u) \in \mathcal{M}_{\text{Nach}}, g_1 \in \mathcal{G}, \tau \in \text{Desc}_{H_{\text{Sub}}}(g_1) \cap \mathcal{J}_{\tau} \\ & \nexists (k_1, k_2) \in \mathcal{M}_{\text{Nach}}, k_1 \in \mathcal{W}_{H_{\text{Sub}}}^{g_1 \rightarrow \tau}, k_2 \in \mathcal{K}, \\ & \exists (v, g_2) \in \mathcal{M}_{\text{Nach}}, v \in \text{Desc}_{H_{\text{Sub}}}(g_1), g_2 = Pa(\tau) \\ \\ 1 - \frac{u \cdot du - 1}{u \cdot du}, & \text{falls } u \neq v, \{u, v\} \subseteq \mathcal{J}, \\ & \exists (v, g_1) \in \mathcal{M}_{\text{Nach}}, g_1 \in \mathcal{G}, \tau \in \text{Desc}_{H_{\text{Sub}}}(g_1), \\ & \nexists (k_1, k_2) \in \mathcal{M}_{\text{Nach}}, k_2 \in \mathcal{W}_{H_{\text{Sub}}}^{g_1 \rightarrow \tau}, k_1 \in \mathcal{K} \\ & \exists (g_2, u) \in \mathcal{M}_{\text{Nach}}, g_2 = Pa(\tau), u \in \text{Desc}_{H_{\text{Sub}}}(g_1) \\ \\ 0, & \text{sonst} \end{cases} \tag{5.47}$$

Eine weitere Verschaltungsmöglichkeit behandelt der vierte Fall: Sind zwei elementare Aktionen u und v über „Umwege“ sequentiell miteinander verschaltet, d.h. liegt v im Inneren eines auf u folgenden selektiven oder iterativen Verschaltungsblocks, dann gibt es im FHIR-Plan keine direkte Kante zwischen u und v . Vielmehr ist in diesem Fall u sequentiell mit einer gruppierenden Aktion $g \in \mathcal{G}$ verschaltet. Es existiert also eine Kante $(g, u) \in \mathcal{M}_{\text{Nach}}$, die angibt, dass auf u eine gruppierende Aktion g folgt. Ist eine elementare Aktion v eine (direkte oder indirekte) Subaktion der Aktion g – also Teil der Menge $\text{Desc}_{H_{\text{Sub}}}(g)$ – so ist ein Übergang von u nach v möglich. Allerdings nur, sofern jeder Knoten k_1 , der zwischen u und v liegt, nicht erst nach dem Ausführen einer beliebigen anderen Aktion k_2 aus der Knotenmenge \mathcal{K} des FHIR-Plans gestartet werden kann. Die Wahrscheinlichkeit von u nach v überzugehen, ist dabei gegeben aus dem Produkt der Wahrscheinlichkeit u zu verlassen, sowie der Wahrscheinlichkeit in v überzugehen.

Der fünfte Fall ist ähnlich zum vierten Fall, allerdings liegt nun u im Inneren eines selektiven oder iterativen Verschaltungsblocks, auf den schließlich v folgt. Auch hier gibt es im FHIR-Plan keine direkte Kante von u nach v . Stattdessen gibt es eine gruppierende Aktion g , auf die die Aktion v folgt. Die Aktion u ist dabei eine (direkte oder indirekte) Subaktion der Aktion g – also auch hier wieder Teil der Menge $\text{Desc}_{H_{\text{Sub}}}(g)$. Ein Übergang von u nach v ist nur möglich, sofern jeder Knoten k_2 , der zwischen u und v liegt, keine Nachfolgeaktionen k_1 hat, die nach seinem Beenden noch ausgeführt werden müssen.

Schließlich wird in Fall fünf, sechs und sieben noch der Spezialfall der stillen Aktionen abgedeckt: Auf eine elementare Aktion u folgt – direkt oder über Umwege – zunächst eine stille Aktion τ und erst dann die elementare Aktion v . In Fall fünf sind u , τ und v direkt sequentiell verschaltet und τ wird übersprungen.

In Fall sechs sind u und τ über Umwege verschaltet, d.h. τ liegt in einem auf u folgenden selektiven oder iterativen Verschaltungsblock. In diesem Fall gibt es eine zwischengeschaltete gruppierende Aktion g_1 . Die Stille Aktion τ ist eine (direkte oder indirekte) Subaktion der Aktion g_1 . Sie ist also Teil der Nachfahren von g_1 und gleichzeitig per Definition Teil der Menge aller

stillen Aktionen \mathcal{J}_τ . Weiterhin darf – vgl. Fall vier – keine Aktion k_1 zwischen g_1 und τ existieren, die erst nach dem Ausführen einer beliebigen anderen Aktion k_2 gestartet werden kann. Es gilt nun, keinen Übergang von der Aktion u in eine stille Aktion τ zu erlauben und diese zu überspringen. Dazu muss ein Nachfolgeknoten v von g_1 gefunden werden, welcher sequentiell mit dem Elternknoten g_2 der stillen Aktion τ verschaltet ist.

Fall sieben ist ähnlich zu Fall sechs, allerdings liegt u nun im Inneren eines selektiven oder iterativen Verschaltungsblockes, auf den schließlich eine stille Aktion τ folgt. Demnach gibt es im FHIR-Plan keine direkte Kante von u nach τ . Stattdessen existiert eine gruppierende Aktion g_1 und die stille Aktion τ ist eine direkte oder indirekte Subaktion von g_1 . Weiterhin darf – vgl. Fall fünf – keine Aktion k_2 zwischen g_1 und τ liegen, die Nachfolgeaktionen k_1 hat, die nach dem Beenden von k_2 noch ausgeführt werden müssen. Es gilt nun wieder, keinen Übergang von u in die stille Aktion τ zu erlauben und diese zu überspringen. Dazu muss ein Nachfolgeknoten g_2 von u gefunden werden, welcher Elternknoten von τ ist und gleichzeitig direkte oder indirekte Subaktion von g_1 .

In allen anderen Fällen ist kein Übergang von u nach v möglich und die Übergangswahrscheinlichkeit wird auf Null gesetzt.

Die Datenoutputs des FHIR-Plans bezeichnen die Merkmale, deren Merkmalswerte während der Operation über die Zeit beobachtet werden. Einige der Datenoutputs $o \in \mathcal{O}_{\text{Out}}$ sind mit verschiedenen Aktionen verbunden, repräsentieren aber dasselbe Merkmal, d. h. sie haben denselben Namen. Sei $\mathcal{O}_{\text{Name}}$ die Menge der eindeutigen Namen der Datenoutputs, die im FHIR-Plan vorhanden sind. Die surjektive Funktion $h_{\text{Nam}} : \mathcal{O} \rightarrow \mathcal{O}_{\text{Name}}$ gebe den Namen eines Datenoutputs o zurück. Weiterhin sei eine zweite, bijektive Funktion $h_{\text{Num}} : \mathcal{O}_{\text{Name}} \rightarrow \{0, \dots, n - 1\}$, mit $n = |\mathcal{O}_{\text{Name}}|$ gegeben. Diese Funktion h_{Num} ordnet jedem Element von $\mathcal{O}_{\text{Name}}$ lexikographisch eine eindeutige natürliche Zahl zu.

Um die zugrunde liegende CPD für jedes Merkmal zu spezifizieren, definiere eine Matrix \mathbf{B}^k für jedes Element $o_{\text{Name}} \in \mathcal{O}_{\text{Name}}$:

$$\exists! \mathbf{B}^k \forall o_{\text{Name}} \in \mathcal{O}_{\text{Name}} : k = h_{\text{Num}}(o_{\text{Name}}).$$

Die Koeffizienten $b_{i,j+1}^k$ jeder Matrix \mathbf{B}^k seien dabei gegeben durch:

$$b_{i,j+1}^k = \begin{cases} P(\underline{X}_k^t = j | \underline{X}_0^t = i), & \text{falls } j = 1 \\ 1 - P(\underline{X}_k^t = j | \underline{X}_0^t = i), & \text{sonst,} \end{cases}$$

wobei $j \in \text{Val}(\underline{X}_k^t) = \{0, 1\}$ und $i \in \text{Val}(\underline{X}_0^t) = \{h_{\text{N}}(v) \mid v \in \mathcal{J}\}$ wie zuvor definiert.

Die bedingten Wahrscheinlichkeiten $P(\underline{X}_k^t = j | \underline{X}_0^t = i)$ werden aus den Datenoutputs abgeleitet. Jedem Datenoutput $o \in \mathcal{O}_{\text{Out}}$ ist ein Wahrscheinlichkeitswert $o.\text{prob}$ zugeordnet und es gilt:

$$P(\underline{X}_k^t = 1 | \underline{X}_0^t = i) = \begin{cases} o.\text{prob}, & \text{falls } \exists(a, o) \in \mathcal{M}_{\text{Out}} \\ & \text{mit} \\ & a \in \mathcal{J}, o \in \mathcal{O}_{\text{Out}}, \\ & (h_{\text{Num}} \circ h_{\text{Nam}})(o) = k, \\ & h_{\text{N}}(a) = i \\ \theta, & \text{sonst.} \end{cases}$$

Hierbei gebe $o.\text{prob}$ die Wahrscheinlichkeit für das Auftreten von o beim Ausführen der Aktion a an. Diese Wahrscheinlichkeit ist im Attribut *probObs.value* des Datenoutputs gespeichert.

Auf diese Weise kann ein FHIR-Plan einer chirurgischen Intervention in ein DBN übersetzt werden, wobei die Parameter von π als Startverteilung, die Parameter von \mathbf{A} für Zustandsübergänge und die Matrizen \mathbf{B}^k für die Wahrscheinlichkeitsverteilungen der Beobachtungen herangezogen werden.

5.5 Vorstudie

Zur praktischen Bewertung des in Kapitel 5.3 dargestellten Standes der Technik wurde das Anwendungsbeispiel einer vereinfachten, nachgestellten Versorgung mit einer Hüft-TEP (vgl. Kapittel 5.1.1) betrachtet.

5.5.1 Material und Methoden

Abbildung 5.19 zeigt eine Momentaufnahme des Experimentallabors aus der Vogelperspektive. Die vorgenommenen Vereinfachungen betreffen die Anzahl der Akteure und die Detaillierungsgrad der nachgestellten Phasen. Der Fokus des nachgestellten Eingriffs liegt damit auf den charakteristischen Aspekten einer realen Operation (vgl. Kapitel 5.1.1), beispielsweise die typische Positionsänderung des Operationsteams.

Die verwendete Sensorbasis besteht aus vier Kinect V2 Tiefenkameras, welche die Distanzwerte über ein Laufzeitverfahren (engl. Time-of-Flight) schätzen [Sel14, Fan15]. Aus den verwendeten Algorithmen zur Erfassung der Skelette der Akteure [Bey16] wird die Positionierung abgeleitet (vgl. P_0 und P_1 in Abbildung 5.19) – weiterhin wird die Anzahl an Personen und die Anzahl der verwendeten Instrumente erfasst. Für Letztere wird das ARTTrack2 System verwendet [Phi16c, Bey16]. Um die erfassten Daten gebündelt bereitzustellen wird die auf dem Robot Operating System (ROS) basierende Forschungsplattform OP:Sense eingesetzt [Bih16].

Der Aufbau liefert 12 verschiedene Lernstichproben $\mathcal{D}_0, \dots, \mathcal{D}_{11}$ des 7 phasigen Arbeitsablaufes (vgl. Phasen in Tabelle 5.5). Jede Lernstichprobe \mathcal{D} besteht aus einer Beobachtungssequenz von Merkmalsvektoren $\mathbf{x}^{0:t}$. Jeder Merkmalsvektor $\mathbf{x} = (x_0, x_1, x_2)^T$ ist gegeben durch die drei Werte der Merkmale Positionierung des Operationsteams, Anzahl Personen am Operationstisch und Anzahl benutzter Instrumente. Die typischen Werte jedes Merkmals über den 7 Phasen ist in Tabelle 5.5 vereinfachend dargestellt. Die tatsächliche Werte der Lernstichproben weichen von dieser idealisierten Darstellung ab, unter anderem wegen auftretenden Verdeckungen von Personen und Gegenständen während der Aufnahmen.

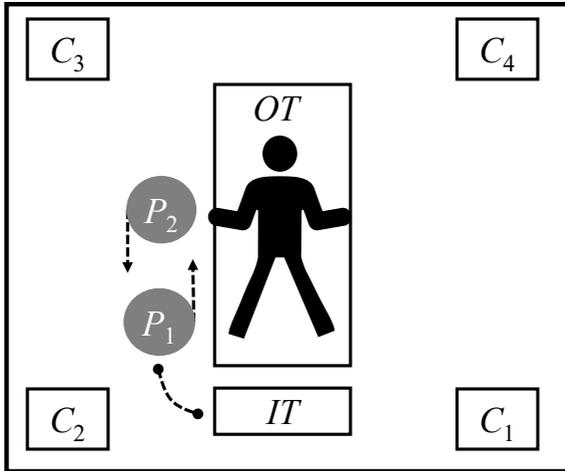


Abbildung 5.19: Die Skizze zeigt die im Rahmen der Vorstudie nachgestellte, vereinfachte endprothetische Versorgung eines Patienten mittels einer Hüft-TEP aus der Vogelperspektive. C_0 bis C_3 repräsentieren die Sensorbasis in Form von vier Tiefenkameras, welche an der Decke des Experimentallabors befestigt sind. Der Instrumententisch (IT) wird genutzt um Instrumente abzulegen, die momentan nicht benötigt werden (gestrichelte Linie). Das Operationsteam besteht aus Person P_0 und Person P_1 . Diese stehen zu Beginn der Operation in der Normalposition bezüglich des Operationstisches (OT), also kaudal des Patienten. Während des Eingriffs wechseln P_0 und P_1 ihre Positionen zweimal (vgl. gestrichelte Pfeile). Modifiziert nach [Phi16c].

Um ein überwachtes Lernen zu ermöglichen, wurden alle Merkmalsvektoren während der Erfassung den entsprechenden vorliegenden Phasen zugeordnet. Das bedeutet die wahre Klasse $\omega(\mathbf{x}) \in \{a, \dots, g\}$ ist für alle Elemente in $\mathcal{D}_0, \dots, \mathcal{D}_{11}$ bekannt. Um zur Laufzeit die aktuell vorliegende Phase zu schätzen (vgl. Kapitel 5.2, Konstituenten des Problems) wird ein RF (vgl. Kapitel 5.3.1) und ein DBN (vgl. Kapitel 5.3.4) eingesetzt. Bei letzterem ist die Struktur des Netzes so gewählt, dass die Wurzel die aktuell vorliegende Phase repräsentiert. Unter Annahme einer geschlossenen Welt summieren sich diese einzelnen Wahrscheinlichkeiten zu 1, da eine der 7 definierten Phasen jeweils

Tabelle 5.5: Die Tabelle zeigt in der ersten Spalte die drei verwendeten Merkmale, welche für die Klassifikation herangezogen wurden. Die zweite Spalte zeigt die dazugehörigen Merkmalswerte. Die darauffolgenden Spalten sind mit einem „x“ markiert gdw. der Merkmalswert typischerweise in der betreffenden Phase (a-g) beobachtet wird. Leicht modifiziert nach [Phi16c].

Merkmal	Wert	Phasen						
		a	b	c	d	e	f	g
Positionierung des Operationsteams	kaudal	x	x	x			x	x
	kranial				x	x		
Anzahl Personen am Operationstisch	1	x						x
	2		x	x	x	x	x	
Anzahl benutzter Instrumente	0	x	x		x		x	x
	1			x				
	2					x		

vorliegen muss. Die Blätter der Struktur des DBNs sind verknüpft mit den geschätzten bedingten Wahrscheinlichkeiten einen bestimmten Merkmalsvektor zu beobachten, gegeben eine Phase. Das bedeutet also $\hat{P}(\mathbf{x}|E)$. Zeitliche Aspekte werden repräsentiert, indem aufeinanderfolgende Zeitschritte miteinander verknüpft werden (vgl. Kapitel 2.4.5). Übergangswahrscheinlichkeiten geben dabei an, wie wahrscheinlich es ist von einer Phase in eine andere Phase überzugehen.

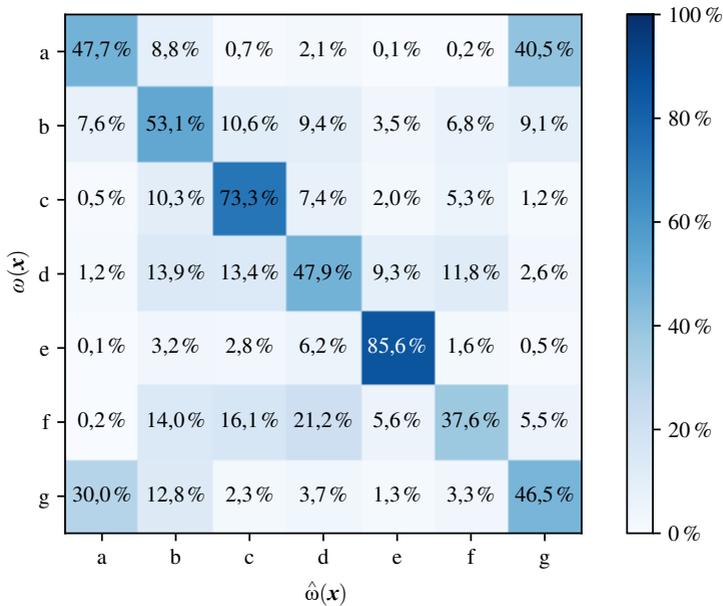


Abbildung 5.20: Normalisierte Konfusionsmatrix der Klassifikationsergebnisse des RF. Eine Zeile zeigt die wahre Klassenzugehörigkeit eines Merkmalsvektors, während eine Spalte die geschätzte Klassenzugehörigkeit widerspiegelt. Folglich repräsentieren die Diagonalelemente den Anteil der korrekt klassifizierten Merkmalsvektoren. Die Konfusion der Matrix zeigt sich in den falsch klassifizierten Elementen abseits der Diagonale, da diese irrtümlich einer anderen (falschen) Klasse zugewiesen wurden. Die Abbildung zeigt eine markante Konfusion bei der RF-Klassifikation zwischen den Phasen a und g. Auch zeigen sich einige Vorhersagen, die sich weit ab der eigentlich korrekten Phasen bewegen. Überarbeitet aus [Sch16].

5.5.2 Ergebnisse

Um die Klassifikatoren zu bewerten wird eine 12-fache Kreuzvalidierung mit den Lernstichproben $\mathcal{D}_0, \dots, \mathcal{D}_{11}$ vollzogen. Dabei fungiert in den 12 Durchläufen jeweils eine andere Stichprobe als Teststichprobe (vgl. Kapitel 5.2.2).

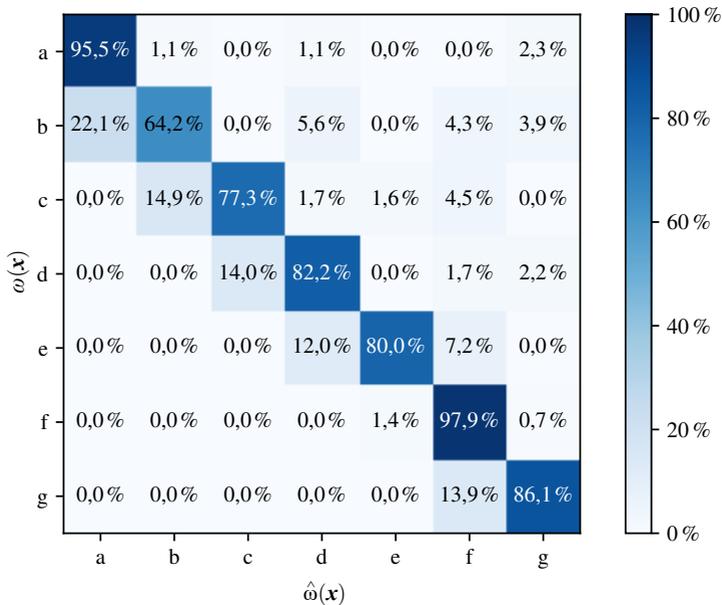


Abbildung 5.21: Normalisierte Konfusionsmatrix der Klassifikationsergebnisse des DBN. Eine Zeile zeigt die wahre Klassenzugehörigkeit eines Merkmalsvektors, während eine Spalte die geschätzte Klassenzugehörigkeit widerspiegelt. In der Abbildung zeigt sich deutlich die überwiegend korrekte Vorhersage der Klassen durch das DBN-Modell ab, die durch eine Konfusion zwischen benachbarten Phasen flankiert wird. Überarbeitet aus [Phi16c].

Um das Verhalten der Modelle genauer zu analysieren, werden die gemittelten Resultate in Form von normalisierten Konfusionsmatrizen dargestellt (vgl. Abbildungen 5.20 und 5.21). Hierbei repräsentieren die Zeilen die wahre Klassenzugehörigkeit $\omega(\mathbf{x})$ über der geschätzten Klassenzugehörigkeit $\hat{\omega}(\mathbf{x})$ in den Spalten. Das bedeutet auf den Diagonalelementen der Matrix findet sich der Anteil jener Merkmalsvektoren wieder, bei denen die durch den Klassifikator geschätzte Klassenzugehörigkeit der wahren Klassenzugehörigkeit entspricht. Falsch klassifizierte Merkmalsvektoren finden sich somit auf den Einträgen abseits der Diagonale.

Hier zeigt sich ein interessanter Unterschied bezüglich der verwendeten Modelle. Während im Falle des RFs sich ähnelnde Phasen oft verwechselt werden ist bei dem DBN Modell wenig Konfusion zu beobachten (vgl. Einträge der Konfusionsmatrizen in den Abbildungen 5.20 und 5.21 der Phasen a und g). Dies ist letztlich dadurch zu erklären, dass das DBN Modell das Wissen nutzt, welches dem Arbeitsablauf der vereinfachten, nachgestellten Hüft-TEP zugrunde liegt: Unsinnige Abfolgen von Phasen werden durch niedrige Übergangswahrscheinlichkeiten verhindert. Die Sequenzmodellierung der DBN reduziert also die Konfusion bei phasenabhängigen Merkmalswerten.

Abbildung 5.20 zeigt weiterhin, dass die Fehlklassifikationen bei dem DBN Modell vorwiegend zwischen benachbarten Phasen auftreten, also beim Wechsel von einer Phase in die nächste. Ausgelöst wird dies durch die Glättung des Klassifikationsergebnisses bei zur Laufzeit wechselnden Beobachtungen. In der Praxis entsteht dadurch ein gewisser zeitlicher Verzug bei der Erkennung eines Phasenübergangs.

Wichtig hierbei ist allerdings, dass dabei nur die bis zum jeweiligen Zeitpunkt vorliegenden Merkmalsvektoren herangezogen werden, also online klassifiziert werden kann (vgl. Kapitel 5.2). Für den gegebenen Anwendungsfall (vgl. Kapitel 5.2) sind diese Fehlklassifikationen weniger von Bedeutung als unsinnige, weit vom eigentlichen Fortschritt des Ablaufs entfernte Phasen vorherzusagen (vgl. Abbildung 5.20). Dennoch gibt es noch weitere Möglichkeiten zur Verbesserung der Ergebnisse.

5.6 Realstudie

Mit Hilfe medizinischer Domänenexperten wurde ausgehend von der Vorstudie eine verkürzte reale Hüft-TEP als sogenanntes Phasenexperiment nachgestellt. Zum Einsatz kommt ein OP-Team in Vollbesetzung und ein biomechanisches Patientenphantom. Details finden sich in den folgenden Unterkapiteln 5.6.1 bis 5.6.2.

5.6.1 Material und Methoden

Die Abbildungen 5.22 und 5.23 zeigt eine Momentaufnahme der Realstudie aus der Vogelperspektive. Das Operationsteam der Realstudie besteht aus dem Operateur *O*, dem ersten Assistenten *1A*, dem zweiten Assistenten *2A* sowie dem Instrumenteur *I*. Damit ist das Operationsteam in Vollbesetzung (vgl. Kapitel 5.1.1).

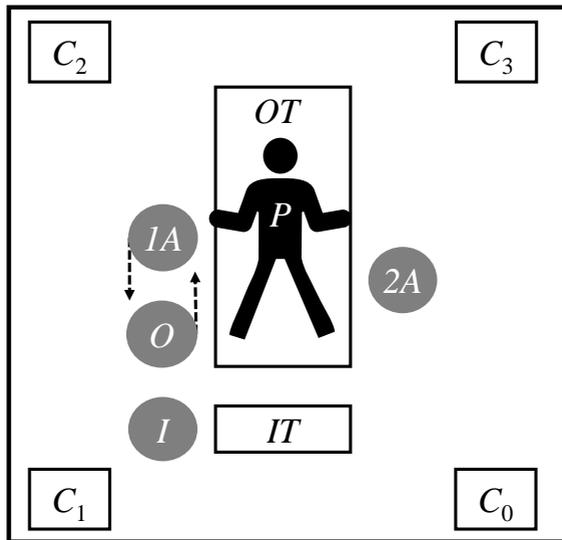


Abbildung 5.22: Die Abbildung zeigt aus der Vogelperspektive die endoprothetische Versorgung mit einer totalen Endorprothese der Hüfte (Hüft-TEP), die im Rahmen der Realstudie mit medizinischen Experten und einem Patientenphantom nachgestellt wurde (vgl. Kapitel 5.1.1). C_0 bis C_3 repräsentieren dabei die Sensorbasis in Form von vier Tiefenkameras, welche an der Decke des Experimentallabors angebracht sind. Die instrumentierende Person *I* reicht Instrumente von Instrumententisch *IT* an den Operateur *O* oder den ersten Assistenten *1A* weiter bzw. nimmt diese wieder entgegen. Letztgenannte stehen zu Beginn der Realstudie in der Normalposition bezüglich des Operationstisches (*OT*), also kaudal des auf dem Rücken liegenden biomechanischen Patientenphantoms *P*. Der zweite Assistent *2A* steht den Kollegen gegenüber und verrichtet während der Intervention vor allem Haltearbeit (vgl. Kapitel 5.1.1). In Anlehnung an [Phi16c, Sch16].

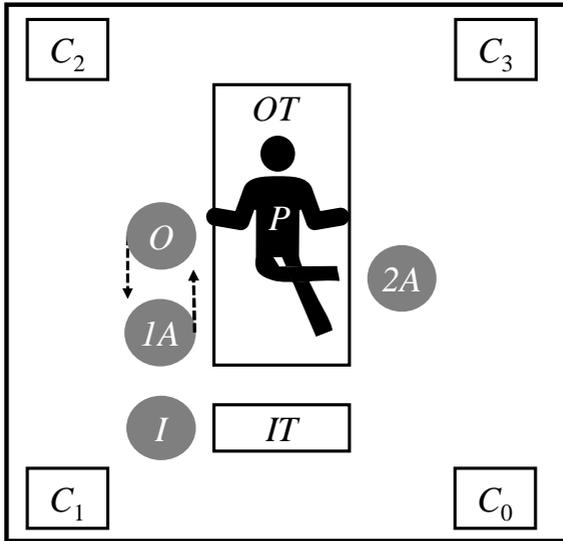


Abbildung 5.23: Die Skizze zeigt wie Abbildung 5.22 die nachgestellte Hüft-TEP in vereinfachter Weise aus der Vogelperspektive. Allerdings befinden sich die Personen *O* und *IA* in gedrehter Positionierung. D. h. Operateur *O* steht kranial des Patientenphantoms um den Schaft des Beines mit dem Hammer bearbeiten zu können.

Das verwendete biomechanische Patientenphantom *P* bildet die knöchernen Strukturen als auch die funktionellen, weichteiligen Strukturen im Bereich des rechten Beines und der rechten Hüfte ab. Dies ermöglicht eine vollständige und realitätsnahe Abarbeitung der betrachteten Operationsphasen von Pfannen- bis zur Schaftpräparation (vgl. Kapitel 5.1.1).

Die verwendete Sensorbasis besteht aus vier Kinect V2 Tiefenkameras, welche die Distanzwerte über ein Laufzeitverfahren (engl. Time-of-Flight) schätzen [Sel14, Fan15]. Um die erfassten Daten gebündelt bereitzustellen wird die auf dem ROS basierende Forschungsplattform OP:Sense eingesetzt [Bih16]. Aus den verwendeten Algorithmen zur Schätzung der Skelette der Akteure

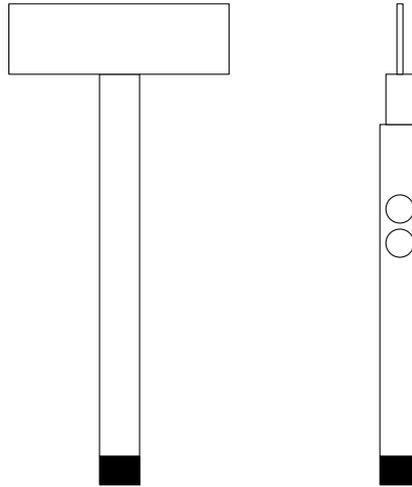


Abbildung 5.24: Die Abbildung zeigt skizzenhaft die chirurgischen Instrumente Hammer (links) und Elektrokauter (rechts). Der Hammer wird üblicherweise in Kombination mit medizinischen Meißeln und Raspeln eingesetzt. Erstere sind Metalldorne wie aus dem Handwerksbereich bekannt. Letztere sind Feilen ähnlich, die allerdings keinen Handgriff, sondern wie ein Meißel, ein Schlagende aufweisen. Der Elektrokauter wird üblicherweise zum Schneiden von Gewebe eingesetzt.

wird deren Positionierung abgeleitet [Bey16]. Weiterhin wird die Verwendung des chirurgischen Instruments Hammer und Elektrokauter¹ erfasst. Abbildung 5.24 zeigt skizzenhaft die chirurgischen Instrumente Hammer und Elektrokauter.

Gemeinsam mit den medizinischen Experten wurde der Ablauf der Phasen der endoprothetischen Versorgung mittels dem direkten, lateralen, transglutealen Zugang nach Bauer modelliert (vgl. Kapitel 5.1.1). Dabei wurden sowohl mögliche als auch unmögliche Abfolgen von Phasen, die Dauern der

¹ Hinweis: Wird teils auch als Diathermie bezeichnet. Ein Elektrokauter nutzt elektrischen Strom um Gewebe zu zerstören bzw. zu schneiden (siehe Elektrokoagulation). Durch die starke Hitzeinwirkung können verletzte Gefäße direkt schneidend verschlossen werden, sodass starke Blutungen vermieden werden.

Tabelle 5.6: Die Tabelle zeigt in der ersten Spalte die drei verwendeten Merkmale, welche für die Klassifikation herangezogen wurden. Die zweite Spalte zeigt die dazugehörigen möglichen Merkmalswerte.

Merkmal	Wert
Positionierung des Operationsteams	kaudal
	kranial
Verwendung des Hammers	nein
	ja
Verwendung des Elektroauters	nein
	ja

Phasen und das verwendete Instrumentarium sowie dessen Einsatz spezifiziert. Abbildung 5.25 zeigt den relevanten Ausschnitt aus dem Model mit den vier Phasen Pfannenpräparation, Pfannenimplantation, Übergang zur Schaftpräparation und der Schaftpräparation. Im Vergleich zu einem vollständigen Eingriff (vgl. Kapitel 5.1.1) ist u. a. die Freipräparation des Zugangsweges, die Durchtrennung des Schenkelhalses als auch die Entfernung des Hüftkopfes vernachlässigt worden. Ebenso fallen die Phasen weg, die bis zur Drainage, Wundverschluss und Anlegen eines Verbandes führen.

Insgesamt konnten so die vier charakteristischen Phasen der insgesamt 14 Phasen umfassenden Operation in der Realstudie betrachtet werden. Dabei wurden vier verschiedene Lernstichproben $\mathcal{D}_0, \dots, \mathcal{D}_3$ erfasst. Jede der Lernstichproben \mathcal{D} besteht aus einer Beobachtungssequenz von Merkmalsvektoren $\mathbf{x}^{0:t}$. Jeder Merkmalsvektor $\mathbf{x} = (x_0, x_1, x_2)^T$ ist gegeben durch die drei Werte der Merkmale Positionierung des Operationsteams, Verwendung des Hammers und Verwendung des Elektroauters (vgl. Tabelle 5.6).

Zur Augmentierung der Beobachtungssequenzen wurde Bootstrapping eingesetzt. Bootstrapping ist eine erstmals in [Efr79] vorgestellte Methode zum

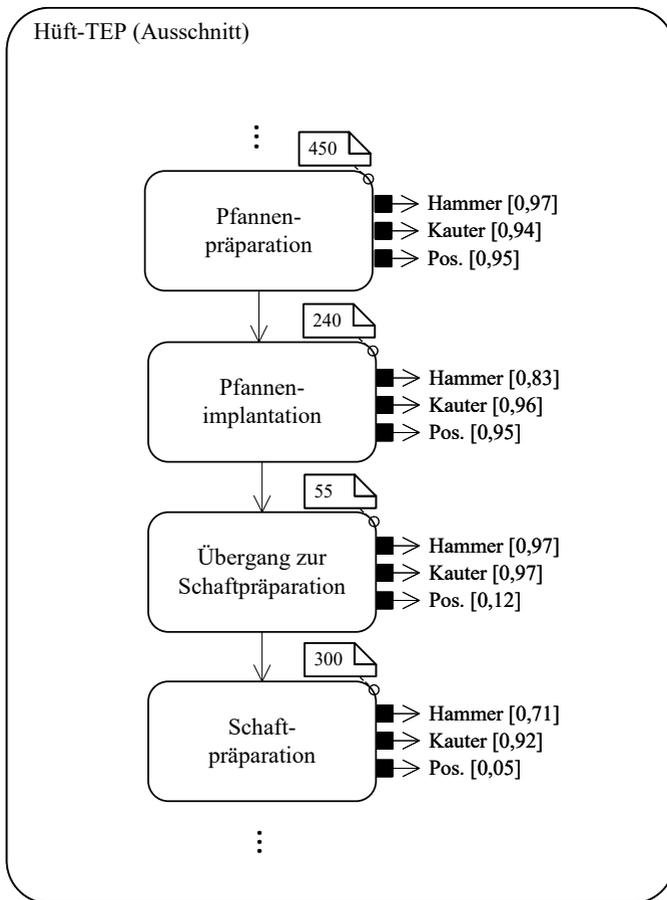


Abbildung 5.25: Die Abbildung zeigt den sequentiellen Ablauf der Realstudie anhand der OP-Phasen fünf bis acht. Diese Phasen beschreiben einen Ausschnitt aus der gesamten OP, welche in Kapitel 5.1.1 ausführlich erläutert wurde. Das Gesamtmodell wurde mit medizinischen Experten entwickelt, welche (un-)mögliche Reihenfolgen der Phasen, die Phasendauern und das verwendete Instrumentarium sowie dessen Einsatz spezifizierten. Die drei Punkte (...) wurden der Abbildung zur Erhöhung der Verständlichkeit hinzugefügt und sollen den Ausschnitt-Charakter der vier Phasen betonen. Die Punkte sind nicht Teil der UML-Spezifikation [OMG17].

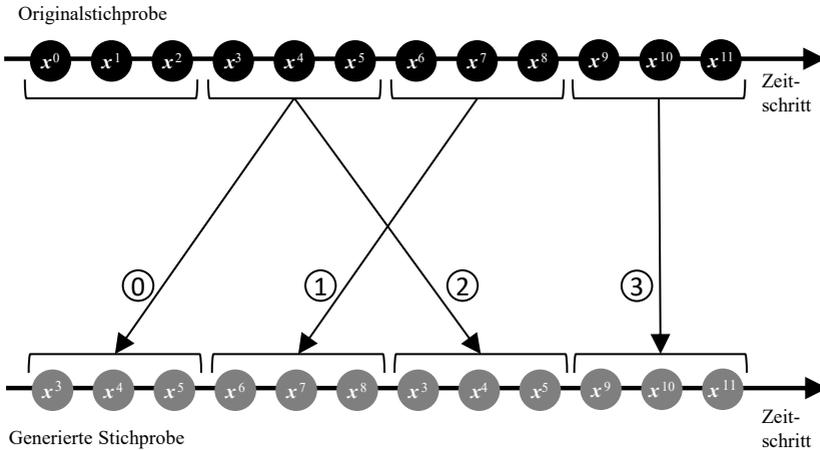


Abbildung 5.26: Die Abbildung zeigt das Vorgehen beim Block Bootstrapping nach [Car86]. Die ursprüngliche Zeitreihe wird hierzu in einem ersten Schritt in Blöcke mit fester Länge unterteilt. Im Beispiel entstehen mit einer Blocklänge von $l = 3$ insgesamt 4 Blöcke. In einem zweiten Schritt wird anschließend aus der Menge der generierten Blöcke vier mal zufällig mit Zurücklegen gezogen. Die so gezogenen Blöcke werden in der Reihenfolge ihres Ziehens zu einer neuen Zeitreihe kombiniert, die die gleiche Länge wie die Originalzeitreihe besitzt.

Resampling, bei der aus einer gegebenen Stichprobe wiederholt Unterstichproben durch zufälliges Ziehen mit Zurücklegen generiert werden. Klassischerweise besteht die Originalstichprobe dabei aus einzelnen, voneinander unabhängigen Beobachtungen.

Im hier betrachteten Fall bestehen die vorliegenden Stichproben $\mathcal{D}_0, \dots, \mathcal{D}_3$ der Realstudie jeweils aus einer Beobachtungssequenz von Merkmalsvektoren $\mathbf{x}^{0:t}$. Weiterhin liegt für diese Sequenzen die Klassenzugehörigkeiten $y^{0:t}$ vor, welche durch die medizinischen Domänenexperten zugeordneten OP-Phasen entsprechen. Insgesamt handelt es sich also um vier multivariate Zeitreihen.

Zur Anwendung auf Zeitreihen muss das in [Efr79] vorgestellte Bootstrap-Verfahren allerdings angepasst werden, da nicht davon ausgegangen werden kann, dass die einzelnen Merkmalsvektoren \mathbf{x}^i mit $i \in \{0, 1, \dots, t\}$ voneinander unabhängige Beobachtungen darstellen. Vielmehr können innerhalb

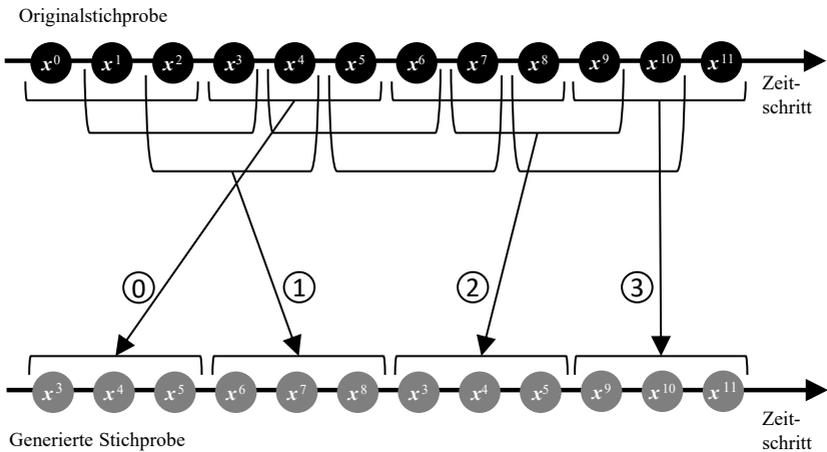


Abbildung 5.27: Die Abbildung zeigt das Vorgehen beim Moving Block Bootstrapping nach [Kun89]. Auch hier wird die ursprüngliche Zeitreihe in Blöcke fester Länge unterteilt, die sich nun allerdings überlappen dürfen. Im Beispiel entstehen mit einer Blocklänge von $l = 3$ insgesamt 10 Blöcke. Aus der Menge dieser 10 Blöcke wird anschließend vier mal zufällig mit Zurücklegen gezogen, und die gezogenen Blöcke in der Reihenfolge ihres Ziehens zu einer neuen Zeitreihe kombiniert.

der Zeitreihe zeitliche Abhängigkeiten auftreten – so kann beispielsweise die Ausprägung eines Merkmalsvektors \mathbf{x}^i von seinen Vorgängern $\{\mathbf{x}^{i-1}, \mathbf{x}^{i-2}, \dots\}$ abhängig sein. Um solche zeitlichen Abhängigkeiten innerhalb von Zeitreihen bewahren zu können, wurde in [Car86] eine angepasste Variante des Bootstrappings, das sogenannte Block Bootstrapping, vorgestellt.

Hierbei wird die vorliegende Zeitreihe $\mathbf{x}^{0:t}$ in einem ersten Schritt zunächst in Blöcke der Länge l , $\{\mathbf{x}^{0:l-1}, \mathbf{x}^{l:2l-1}, \mathbf{x}^{2l:3l-1}, \dots, \mathbf{x}^{t-l:t}\}$, unterteilt. Anschließend werden in einem zweiten Schritt statt einzelner Beobachtungen $\mathbf{x}^i, i \in \{0, 1, \dots, t\}$ ganze Blöcke konsekutiver Beobachtungen mit Zurücklegen gezogen und zu einer neuen Zeitreihe kombiniert. Abbildung 5.26 verdeutlicht das Vorgehen anhand einer beispielhaften Blocklänge von $l = 3$.

Durch dieses Vorgehen können zwar die innerhalb der einzelnen Blöcke bestehenden zeitlichen Abhängigkeiten bewahrt werden, jedoch gehen zeitliche

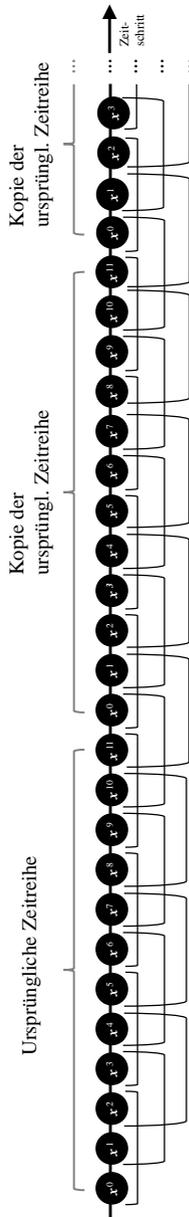


Abbildung 5.28: Die Abbildung zeigt die Vorbereitung der Zeitreihe und Unterteilung in Blöcke beim Circular Bootstrapping nach [Pol91], beispielhaft für eine Blocklänge von $l = 3$. Die ursprüngliche Zeitreihe wird mehrfach in Kopie aneinander gehängt, bevor diese in sich überlappende Blöcke fester Länge unterteilt wird. Aus der Menge der so generierten Blöcke wird anschließend zufällig mit Zurücklegen gezogen. Die gezogenen Blöcke werden in der Reihenfolge ihres Ziehens zu einer neuen Zeitreihe mit der gleichen Länge wie die Originalzeitreihe kombiniert (hier nicht abgebildet).

Abhängigkeiten, die über die Blocklänge l hinaus bestehen, verloren. Entscheidend ist also die Wahl einer für die Stichprobe optimalen Blocklänge l . Diese hängt sowohl von der Länge der vorliegenden Zeitreihe, als auch von deren Autokorrelationsfunktion ab, vgl. [Pol04]. Im hier vorgestellten Ansatz wird die optimale Blocklänge nach dem in [Pol04] und [Pat09] publizierten Algorithmus mithilfe der Toolbox ARCH [She19] bestimmt.

Weiterhin wurden in der Literatur verschiedene Varianten des Block Bootstrappings vorgeschlagen, vgl. [Büh02]. Beispielsweise werden bei dem in Abbildung 5.27 dargestellten Moving Block Bootstrap Verfahren nach [Kun89] zusätzlich sich überlappende Blöcke der Länge l berücksichtigt. Jedoch sind hierdurch in den entstehenden Bootstrap-Stichproben die Merkmalsvektoren, die sich am Anfang und Ende der Originalzeitreihe befanden, systematisch unterrepräsentiert.

Dies versucht man beim Circular Block Bootstrapping nach [Pol91] zu umgehen, indem man die Originalzeitreihe vor dem Resampling wiederholt aneinander reiht und auch hier sich überlappende Blöcke zulässt (vgl. Abbildung 5.28). Weiterhin kann Circular Bootstrapping sowohl für reellwertige, als auch für kategoriale Zeitreihen angewendet werden [Büh02]. Das Verfahren des Circular Bootstrappings wird daher im Folgenden für die Erweiterung der Datenbasis eingesetzt.

Bei den vorliegenden Zeitreihen der Realstudie handelt es sich um Instanziierungen konkreter Merkmalswerte über die Zeit, die dazu herangezogen werden, die OP-Phase über die Zeit zu klassifizieren, vgl. Kapitel 5.2. Damit eine solche Klassifikation funktioniert, ist zu erwarten, dass sich der Erwartungswert und die Varianz der Merkmale über die einzelnen OP-Phasen ändern, bzw. bei nominalen Merkmalswerten, wie beispielsweise der Positionierung, ist denkbar, dass eine Ausprägung eine bestimmte OP-Phase dominiert, sich dies in der nächsten OP-Phase jedoch ändert.

Zum Augmentieren der Datenbasis wird daher Circular Bootstrapping nicht über alle Phasen einer Aufnahme hinweg durchgeführt, sondern die jeweilige Aufnahme vorab in ihre einzelnen Phasen unterteilt und das Bootstrapping je

Phase separat durchgeführt (vgl. Abbildung 5.29). Hierfür wird für jede Phase individuell eine optimale Blocklänge bestimmt.

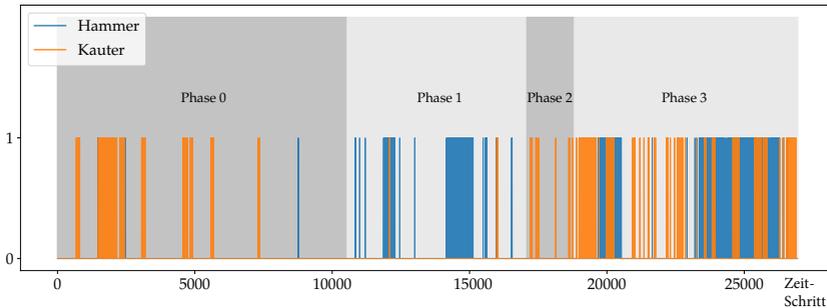


Abbildung 5.29: Beispielhafte Zeitreihe mit Instanziierungen der Merkmalswerte für die Merkmale Hammer und Elektrokauter für die vier OP-Phasen der Realstudie. Für die Augmentierung der Datenbasis werden die Zeitreihen zunächst in die einzelnen OP-Phasen unterteilt. Für jede der Phasen werden separat optimale Blocklängen bestimmt und Circular Block Bootstrapping durchgeführt. Die so generierten Bootstrap-Stichproben der Einzelphasen werden anschließend nach Phasennummer sortiert und zu Gesamtzeitreihen eines Arbeitsablaufs kombiniert.

Abbildung 5.30 zeigt beispielhaft für eine Originalstichprobe die mittels Circular Block Bootstrapping generierten zusätzlichen Abläufe anhand der Merkmale Hammer und Elektrokauter.

5.6.2 Ergebnisse

Mit den insgesamt 100 Stichproben $\mathcal{D}_0^*, \dots, \mathcal{D}_{99}^*$ wurde eine 4-fach Kreuzvalidierung durchgeführt (vgl. Kapitel 5.2.2). In den 4 Durchläufen fungiert jeweils ein anderes Bündel von 25 Stichproben als Testmenge (vgl. Abbildung 5.31). Dabei wird ein RF, ein DBN sowie ein Expertenbasiertes DBN (EBN) miteinander verglichen. Für die Festlegung der Modellparameter des RFs wurde zusätzlich eine 3-fach Kreuzvalidierung eingebettet und mittels Gridsearch die Anzahl der Bäume, Gütekriterium für eine Teilung, Anzahl berücksichtigter

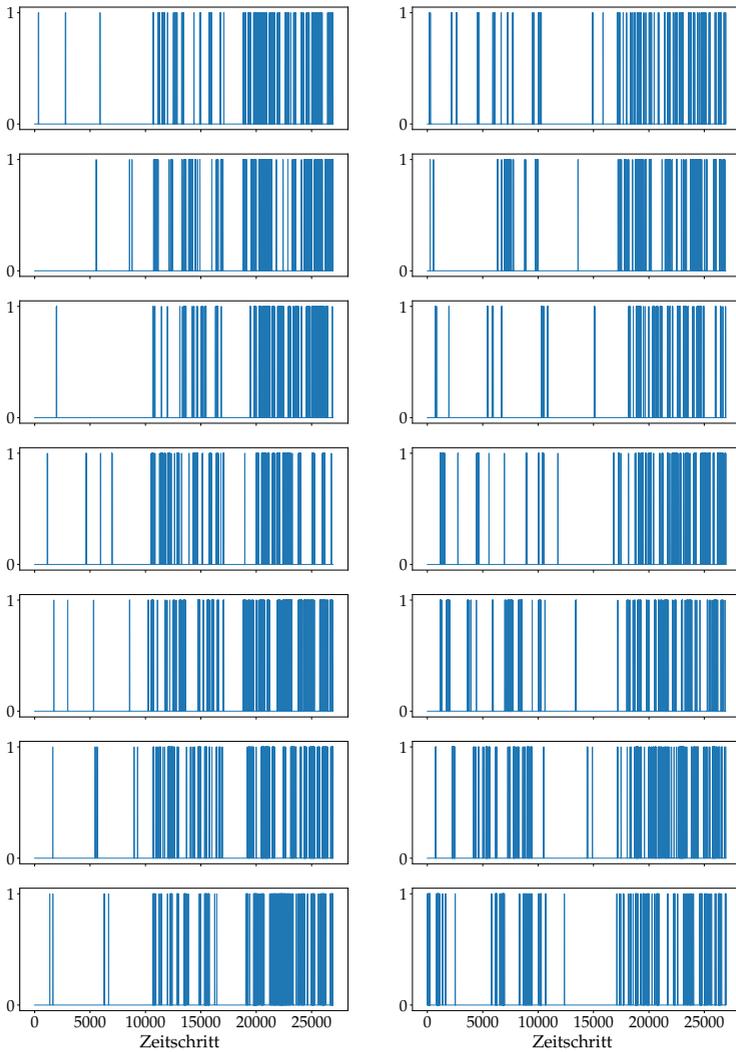


Abbildung 5.30: Beispielhafte durch Circular Block Bootstrapping generierte Zeitreihen für die Originalstichprobe aus Abbildung 5.29. Generierte Zeitreihen für das Merkmal Hammer (links) und das Merkmal Elektrokauter (rechts).

Komponenten des Merkmalsvektors bei Teilung sowie dem Abbruchkriterium über die Mindestanzahl der Merkmalsvektoren pro Blatt variiert (vgl. Kapitel 5.3.1). Das EBN wurde um das Expertenwissen ergänzt, welches die Dauern der Phasen, die Festlegung der Reihenfolge der Phasen und Verwendung von Instrumenten sowie deren Einsatz betrifft. D. h. das EBN ist im Hinblick auf $P(\underline{X}_0^0)$ und $P(\underline{X}_0^t | \underline{X}_0^{t-1})$ modifiziert; unmögliche Übergänge sind beispielsweise mit einer Übergangswahrscheinlichkeit von 0 gesetzt. Für diese Komponente der Wissensakquisition (vgl. Kapitel 2.2) wurde das UML Aktivitätsdiagramm aus Abbildung 5.25 auf Seite 173 eingesetzt.

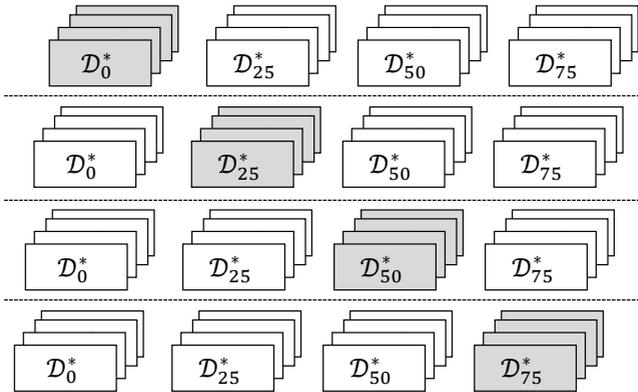


Abbildung 5.31: Die Abbildung zeigt die Aufteilung der insgesamt 100 Lernstichproben $\mathcal{D}_0^*, \dots, \mathcal{D}_{99}^*$. Es werden vier Durchläufe durchgeführt (siehe vier Zeilen). In jedem Durchlauf fungiert ein Paket von jeweils 25 Beobachtungssequenzen als Testmenge (grau). Jedes der Pakete wurde aus einer der originalen Beobachtungssequenzen $\mathcal{D}_0, \dots, \mathcal{D}_3$ gewonnen.

In den Tabellen 5.7 bis 5.10 finden sich die Auswertungen der betrachteten Modelle. In den ersten Zeilen sind RF, DBN und EBN mit den entsprechenden Kennwerten sowie empirischer Standardabweichung aufgelistet (vgl. Kapitel 5.2.2). Dabei wird den Modellen sukzessive pro Zeitschritt ein weiterer Merkmalsvektor zur Verfügung gestellt – die Beobachtungssequenz baut sich also

Tabelle 5.7: Die Tabelle zeigt in der ersten Spalte die betrachteten Klassifikatormodelle und in den folgenden vier Spalten die Kennwerte zur Bewertung der Problemlösung (vgl. Kapitel 5.2.2) und deren empirische Standardabweichung ($\pm x\%$). Bezüglich der *Precision* zeigt sich eine schwache Leistung des RF. Weiterhin ist die Leistung des EBN in allen Durchläufen signifikant besser als das rein datenbasiert trainierte DBN. Ersteres ist sehr nah an der bestmöglichen Klassifikation für den gegebenen Anwendungsfall.

Precision der einzelnen Durchläufe der Kreuzvalidierung

Modell	0	1	2	3
RF	74,2 % $\pm 0,57\%$	70,0 % $\pm 0,73\%$	74,1 % $\pm 0,55\%$	68,1 % $\pm 0,78\%$
DBN	81,8 % $\pm 1,6\%$	79,1 % $\pm 2,5\%$	85,3 % $\pm 1,8\%$	80,9 % $\pm 2,0\%$
EBN	95,5 % $\pm 3,9\%$	94,4 % $\pm 3,6\%$	97,6 % $\pm 1,9\%$	95,4 % $\pm 3,4\%$
Viterbi	96,6 % $\pm 3,0\%$	96,1 % $\pm 2,8\%$	98,2 % $\pm 1,9\%$	97,1 % $\pm 2,7\%$

über die Zeitschritte hinweg auf und das Modell hat keinen Einblick in zukünftige Merkmalsvektoren (vgl. Kapitel 5.2). Als Referenz ist die best mögliche Klassifikationsbewertung angegeben, die erreicht werden kann, wenn die gesamte Beobachtungssequenz a-priori bekannt ist und der wahrscheinlichste Pfad durchlaufen wird (Viterbi).

In Tabelle 5.7 findet sich die *Precision* über den einzelnen Durchläufen der Kreuzvalidierung. Neben der schwachen Leistung des RF stellt sich das EBN in allen Durchläufen als signifikant besser als das DBN dar.

Betrachtet man den *Recall* in Tabelle 5.8, so ergibt sich ein ähnliches Bild. Auch hier ist Leistung des RF schwach, während das EBN in allen Durchläufen signifikant besser als das DBN ist. Eine Ausnahme stellt allerdings der letzte Durchlauf der Kreuzvalidierung dar; Hier kann das EBN keine signifikant bessere Leistung bezüglich des *Recalls* erbringen.

Tabelle 5.8: Die Tabelle zeigt die betrachteten Klassifikatormodelle und die Kennwerte zur Bewertung der Problemlösung mitsamt deren empirischer Standardabweichung. Der RF zeigt durchweg eine signifikant schwächere Leistung als die Vergleichsmodelle. Weiterhin ist die Leistung des EBN in drei von vier Durchläufen signifikant besser als das rein datenbasiert trainierte DBN.

Recall der einzelnen Durchläufe der Kreuzvalidierung

Modell	0	1	2	3
RF	58,2 % ± 1,7 %	54,9 % ± 2,6 %	60,9 % ± 1,7 %	64,1 % ± 1,7 %
DBN	83,6 % ± 3,3 %	81,8 % ± 4,6 %	90,1 % ± 2,6 %	93,3 % ± 3,0 %
EBN	96,3 % ± 3,2 %	96,0 % ± 3,2 %	98,0 % ± 1,6 %	97,2 % ± 2,5 %
Viterbi	96,7 % ± 3,1 %	97,1 % ± 2,5 %	98,4 % ± 1,8 %	98,0 % ± 2,4 %

Tabelle 5.9: Die Tabelle zeigt in der ersten Spalte die betrachteten Klassifikatormodelle. In den folgenden vier Spalten sind die Kennwerte zur Bewertung der Problemlösung aufgelistet. Darüber hinaus sind jeweils die empirischen Standardabweichungen angegeben. Bezüglich des *Jaccard-Score* (Jaccard) zeigt sich eine schwache Leistung des RF. Weiterhin ist die Leistung des EBN in allen Durchläufen signifikant besser als das rein datenbasiert trainierte DBN.

Jaccard der einzelnen Durchläufe der Kreuzvalidierung

Modell	0	1	2	3
RF	34,4 % ± 1,7 %	30,7 % ± 1,8 %	36,3 % ± 2,0 %	33,6 % ± 2,0 %
DBN	66,8 % ± 4,4 %	65,8 % ± 6,0 %	76,1 % ± 3,9 %	74,4 % ± 4,8 %
EBN	91,9 % ± 6,7 %	90,6 % ± 6,0 %	95,6 % ± 3,1 %	92,7 % ± 5,0 %
Viterbi	93,4 % ± 5,7 %	93,4 % ± 4,5 %	96,7 % ± 3,4 %	95,1 % ± 4,3 %

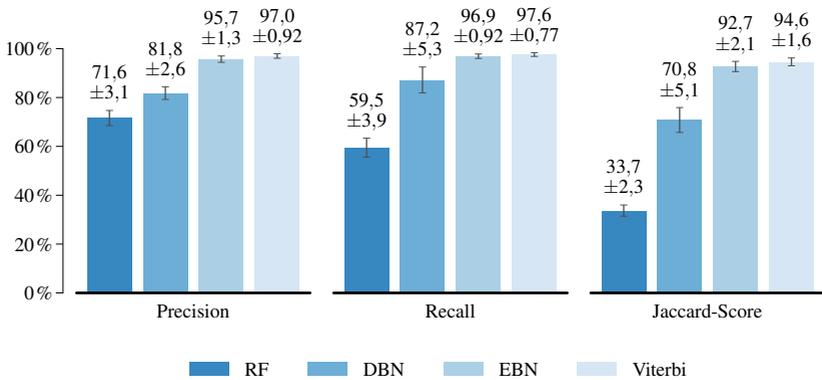


Abbildung 5.32: Die Abbildung zeigt die Gesamtergebnisse der Kreuzvalidierung bezüglich *Precision*, *Recall* und *Jaccard-Score* - also den Kennwerten zur Bewertung der Problemlösung (vgl. Kapitel 5.2.2). Darüber hinaus sind jeweils die empirischen Standardabweichungen angegeben. Dabei zeigt sich der RF als signifikant schlechter als die Vergleichsmodelle. Ebenso zeigt das DBN signifikant schlechtere Ergebnisse als das EBN.

Tabelle 5.10: Die Tabelle zeigt in den Spalten die Gesamtergebnisse der Kreuzvalidierung bezüglich *Precision*, *Recall* und *Jaccard-Score* (Jaccard). Weiterhin sind jeweils die empirischen Standardabweichungen angegeben. Hinweis: Eine alternative Darstellung der Ergebnisse findet sich in Abbildung 5.32.

Modell	Precision	Recall	Jaccard
RF	71,6 % ± 3,1 %	59,5 % ± 3,9 %	33,7 % ± 2,3 %
DBN	81,8 % ± 2,6 %	87,2 % ± 5,3 %	70,8 % ± 5,1 %
EBN	95,7 % ± 1,3 %	96,9 % ± 0,92 %	92,7 % ± 2,1 %
Viterbi	97,0 % ± 0,91 %	97,6 % ± 0,77 %	94,6 % ± 1,6 %

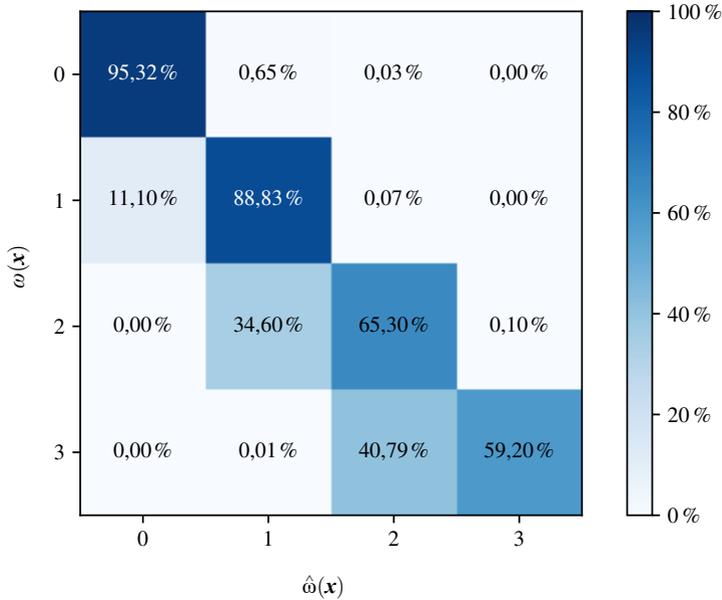


Abbildung 5.33: Normalisierte Konfusionsmatrix der Klassifikationsergebnisse des DBN. Eine Zeile zeigt die wahre Klassenzugehörigkeit eines Merkmalsvektors, während eine Spalte die geschätzte Klassenzugehörigkeit widerspiegelt. In der Abbildung zeichnet sich die optimale Klassifikation mit Werten auf der Winkelhalbierenden ab – es herrscht allerdings eine gewisse Konfusion zwischen benachbarten und weiter entfernten Phasen.

Tabelle 5.9 zeigt den Jaccard-Score¹ über den Durchläufen der Kreuzvalidierung. Zum wiederholten Male zeichnet sich die signifikant bessere Leistung des EBNs ab.

Das Gesamtergebnis der Kreuzvalidierung ist schließlich in Abbildung 5.32 und in Tabelle 5.10 aufgetragen. Dabei liefert der RF durchweg signifikant schlechtere Ergebnisse als die Vergleichsmodelle. Es zeigt sich weiterhin, dass

¹ Hinweis: Erklärungen und Definitionen zur Bewertung der Problemlösung finden sich in Kapitel 5.2.2 ab Seite 117.

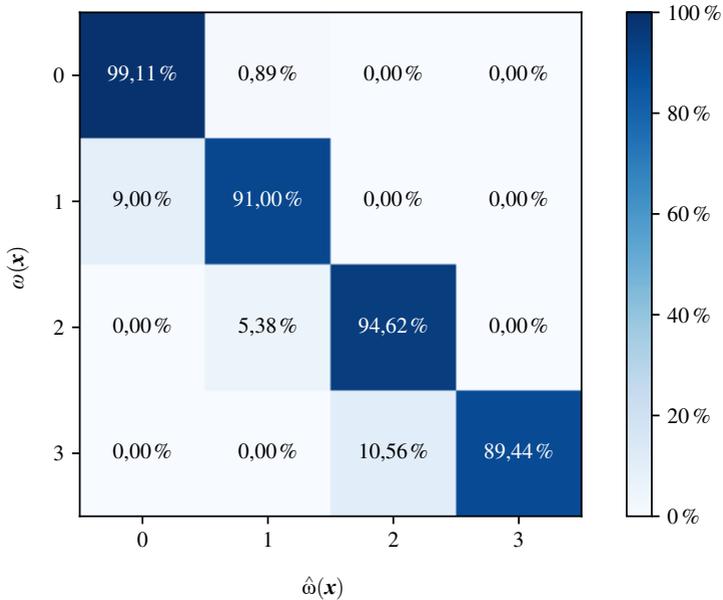


Abbildung 5.34: Normalisierte Konfusionsmatrix der Klassifikationsergebnisse des EBN. Eine Zeile zeigt die wahre Klassenzugehörigkeit eines Merkmalsvektors, während eine Spalte die geschätzte Klassenzugehörigkeit widerspiegelt. In der Abbildung zeigt sich deutlich die überwiegend korrekte Vorhersage der Klassen durch das DBN-Modell ab, die durch eine gewisse Konfusion zwischen benachbarten Phasen flankiert wird.

das EBN bezüglich der gewählten Kennwerte der Leistungsbewertung durchweg signifikant besser ist als das DBN. Darüber hinaus sind die Kennwerte des EBN nahe der best möglichen Klassifikationsbewertung unter Kenntnis der gesamten Beobachtungssequenz (Viterbi).

Genauere Einblicke bezüglich der klassenspezifischen Leistungen liefern die Abbildungen 5.33 und 5.34. Diese zeigen die Konfusionsmatrizen für eine exemplarische Beobachtungssequenz der Testmenge.

Es ist zu erkennen, dass das EBN durch das Ausschließen unsinniger Übergänge eine stärkere Ausprägungen auf der Winkelhalbierenden zeigt als das DBN. Weiterhin zeigt sich beim EBN eine bessere Abgrenzung der Phasen auf den ungesehenen Merkmalsvektoren der Testmenge.

5.7 Galle

Zur praktischen Bewertung des in Kapitel 4 vorgestellten Verfahrens zur Wissensakquisition und Problemlösung wird im Folgenden die laparoskopische Cholezystektomie betrachtet. Hierbei handelt es sich um einen prototypischen Vertreter minimalinvasiver Interventionen mit vergleichsweise häufigen Fallzahlen (vgl. Kapitel 5.1.2).

5.7.1 Material und Methoden

Die laparoskopische Cholezystektomie wird über vier künstliche Körperöffnungen durchgeführt. Durch diese Körperöffnungen wird dann das Instrumentarium eingeführt, welches genutzt wird, um letztlich die Gallenblase¹ zu ektomieren. Bei dem oben genannten Verfahren kommt weiterhin ein Laparoskop² zum Einsatz. Abbildung 5.35 zeigt den Interventionsraum aus der Vogelperspektive.

Für den betrachteten Anwendungsfall liegen 57 vollständige Stichproben solcher Cholezystektomien vor, die von 13 verschiedenen Chirurgen während realer chirurgischer Intervention an Patienten durchgeführt und mittels eines Laparoscops aufgezeichnet wurden [Twi17]. Unvollständige Stichproben wurden entfernt – die verworfenen Aufnahmen enthalten teils obligatorische Phasen nicht, da die Aufnahme beispielsweise zu spät gestartet wurde oder während der Aufnahme die Aufzeichnung unterbrochen wurde.

¹ Hinweis: Engl. Cholecyst. Weitere Details siehe Kapitel 5.1.2.

² Hinweis: Ein spezielles Endoskop, welches für die Durchführung einer Laparoskopie, also einer Bauchspiegelung, verwendet wird.

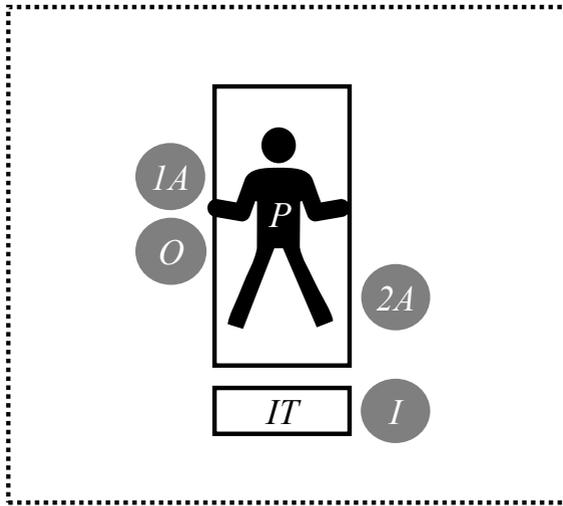


Abbildung 5.35: Die Abbildung zeigt eine vereinfachte Momentaufnahme des Operationsaals bei der Durchführung der laparoskopischen Cholezystektomie. Der Patient P liegt rücklinks auf dem Operationstisch. Operateur O , erster Assistent $1A$, zweiter Assistent $2A$ und der Instrumentierende I stehen um den Tisch herum. Der Instrumentierende I reicht die Instrumente (vgl. Tabelle 5.11) von Instrumentiertisch IT weiter. Die Instrumente werden über künstliche Körperöffnungen in den Patienten P eingeführt. Als Bildgebung fungiert ein Laparoskop.

Jede der vollständigen Stichproben \mathcal{D} besteht aus einer Beobachtungssequenz von Merkmalsvektoren $\mathbf{x}^{0:t}$, wobei jeder Merkmalsvektor $\mathbf{x} = (x_0, x_1, x_2, x_3, x_4, x_5, x_6)^T$ gegeben ist durch die sieben Merkmale die in Tabelle 5.11 gelistet sind. Es handelt sich dabei um das für den Eingriff notwendige Instrumentarium.

Weiterhin ist für jeden Merkmalsvektor dessen Klassenzugehörigkeit $\omega(\mathbf{x}) \in \{0, \dots, 6\}$ über medizinische Experten bekannt. Die Datensätze sollen im Folgenden zur Darstellung der in Kapitel 4 ausgearbeiteten Vorgehensweise verwendet werden.

Tabelle 5.11: Die Tabelle zeigt in der ersten Spalte die sieben verwendeten Merkmale, welche für die Klassifikation der sieben Phasen herangezogen wurden. Die zweite Spalte zeigt die möglichen Merkmalswerte.

Merkmal	Wert
Verwendung der Fasszange	nein ja
Verwendung des Elektrokauters	nein ja
Verwendung der Präparationszange	nein ja
Verwendung der Schere	nein ja
Verwendung des Clipapplikators	nein ja
Verwendung der Spülung	nein ja
Verwendung des Bergebeutel	nein ja

Die chirurgische Intervention der Cholezystektomie kann in sieben Phasen eingeteilt werden. Im Einzelnen sind diese Phasen gemäß den verwendeten Stichproben gegeben als:

0. Präparation vorbereiten
1. Calot'sches Dreieck freipräparieren
2. Gefäße clippen und schneiden
3. Gallenblase freipräparieren
4. Gallenblase in Bergebeutel verbringen
5. Leberbett säubern und koagulieren
6. Bergebeutel herausführen

Zwischen den Phasen bestehen Abhängigkeiten – diese lassen sich anschaulich mittels eines Graphen formalisieren.

Abbildung 5.36 zeigt eine UML-Aktivität als Wissenserwerbskomponente über den zugrunde liegenden Ablauf des betrachteten Anwendungsfalls. Hierbei wurden sowohl mögliche als auch unmögliche Abfolgen von Phasen, die Dauern der Phasen und das verwendete Instrumentarium sowie dessen Einsatz spezifiziert.

5.7.2 Ergebnisse

Die Leistungsbewertung des Klassifikationsproblems bezogen auf die einzelnen Phasen ist den Tabellen 5.12, 5.13 und 5.14 zu entnehmen. In den Zeilen der Tabellen sind die verwendeten Modelle RF, DBN sowie EBN aufgeführt. Als Referenz ist die best mögliche Klassifikationsbewertung angegeben, die erreicht werden kann, wenn die gesamte Beobachtungssequenz a-prior bekannt ist und der wahrscheinlichste Pfad durchlaufen wird (Viterbi).

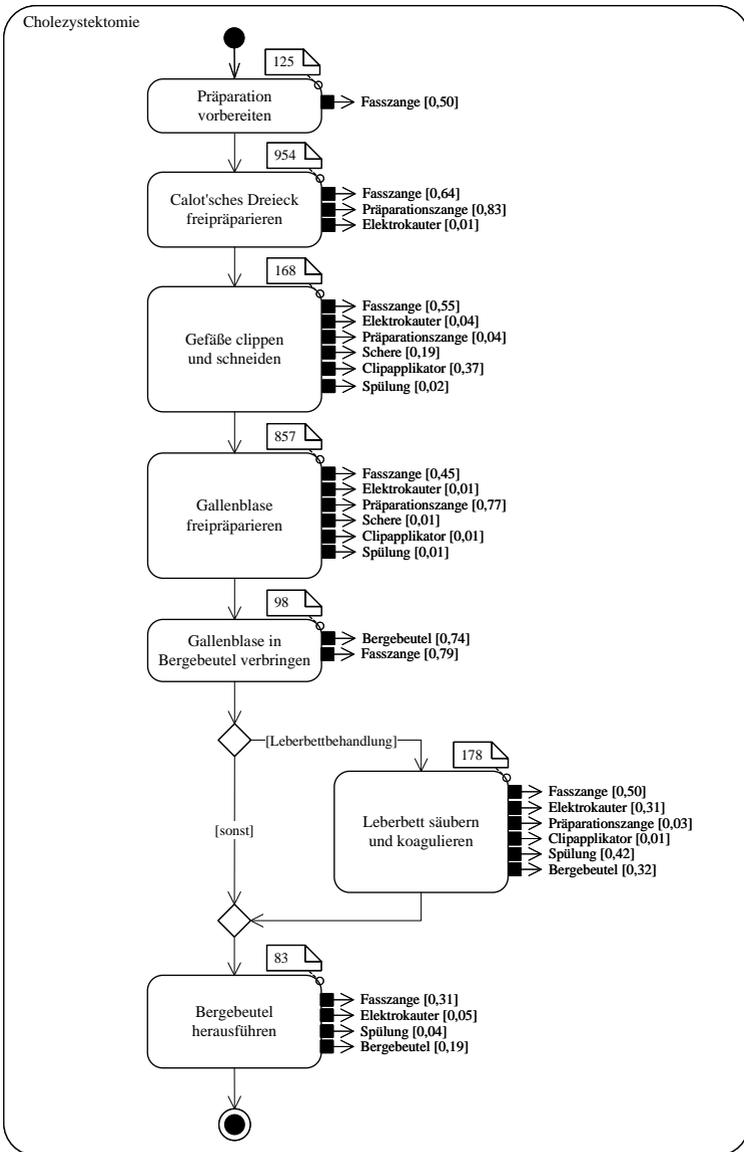


Abbildung 5.36: UML Aktivität des betrachteten Anwendungsfalls der Cholezystektomie.

Tabelle 5.12: Die Tabelle zeigt in der ersten Spalte die verwendeten Modelle, welche für die Klassifikation herangezogen wurden. Die weiteren Spalten zeigen die Bewertung der Problemlösung auf die einzelnen Phasen heruntergebrochen. Dabei sind die dargestellten Kennwerte das arithmetische Mittel aller Durchläufe der Kreuzvalidierung.

<i>Precision über den Phasen</i>							
Modell	0	1	2	3	4	5	6
RF	18,3 % ± 1,5 %	64,7 % ± 0,45 %	94,2 % ± 0,85 %	56,2 % ± 1,1 %	71,3 % ± 2,2 %	58,1 % ± 5,1 %	17,7 % ± 0,89 %
DBN	95,4 % ± 0,95 %	97,3 % ± 2,5 %	54,4 % ± 5,2 %	61,8 % ± 0,89 %	45,5 % ± 2,1 %	53,7 % ± 4,3 %	24,2 % ± 0,73 %
EBN	97,8 % ± 0,24 %	98,4 % ± 1,3 %	70,3 % ± 3,6 %	64,0 % ± 0,54 %	60,9 % ± 3,1 %	52,7 % ± 4,4 %	30,5 % ± 2,0 %
Viterbi	98,6 % ± 2,6 %	98,5 % ± 1,8 %	72,6 % ± 2,1 %	98,1 % ± 1,8 %	73,4 % ± 3,2 %	90,5 % ± 2,1 %	78,3 % ± 2,6 %

Durch das Expertenwissen ergibt sich in einzelnen Phasen eine Verbesserung der Klassifikationsleistung. Tabelle 5.12 zeigt den Sachverhalt für die Precision. In den Phasen 0, 2, 3, 4 und 6 ist das EBN signifikant besser als das rein datenbasiert eingelernte DBN. Umgekehrt zeichnet sich in den Phasen 1 und 5 keine Verbesserung bezüglich der Precision ab.

In Tabelle 5.13 findet sich ein ähnliches Ergebnis für den Recall. Überwiegend ist das EBN signifikant besser als das DBN, lediglich in den Phasen 4, 5 und 6 ergeben sich Abweichungen. Schließlich zeigt der Jaccard-Score in Tabelle 5.14 dass das EBN in allen Phasen signifikant besser ist, ausgenommen der Phase 5 – hier bleibt die Leistung bezüglich des DBNs aber erhalten.

Genauere Einblicke geben die Konfusionsmatrizen in den Abbildungen 5.37 und 5.38. Die Matrizen sind bezüglich ihrer Zeilen (wahren Klassen) normalisiert, d.h. die Einträge der Zeilen ergeben sich zu 100 %.

Tabelle 5.13: Die Tabelle zeigt in der ersten Spalte die verwendeten Modelle, welche für die Klassifikation herangezogen wurden. Die weiteren Spalten zeigen die Bewertung der Problemlösung auf die einzelnen Phasen heruntergebrochen. Dabei sind die dargestellten Kennwerte das arithmetische Mittel aller Durchläufe der Kreuzvalidierung.

<i>Recall</i> über den Phasen							
Modell	0	1	2	3	4	5	6
RF	52,3 % ± 2,1 %	51,6 % ± 0,93 %	65,9 % ± 2,2 %	43,2 % ± 0,78 %	75,6 % ± 1,4 %	53,9 % ± 7,0 %	56,4 % ± 1,6 %
DBN	73,9 % ± 5,9 %	42,5 % ± 1,5 %	47,6 % ± 2,1 %	83,5 % ± 1,1 %	50,9 % ± 1,8 %	58,4 % ± 9,6 %	87,0 % ± 3,6 %
EBN	95,7 % ± 1,8 %	46,4 % ± 0,33 %	76,6 % ± 1,5 %	86,2 % ± 0,74 %	53,4 % ± 1,7 %	60,2 % ± 8,4 %	83,8 % ± 5,2 %
Viterbi	97,1 % ± 0,85 %	91,4 % ± 1,5 %	94,5 % ± 1,8 %	91,9 % ± 1,2 %	92,8 % ± 2,1 %	78,7 % ± 1,9 %	88,0 % ± 1,9 %

Tabelle 5.14: Die Tabelle zeigt in der ersten Spalte die verwendeten Modelle, welche für die Klassifikation herangezogen wurden. Die weiteren Spalten zeigen die Bewertung der Problemlösung auf die einzelnen Phasen heruntergebrochen. Dabei sind die dargestellten Kennwerte das arithmetische Mittel aller Durchläufe der Kreuzvalidierung.

<i>Jaccard-Score</i> über den Phasen							
Modell	0	1	2	3	4	5	6
RF	14,2 % ± 1,4 %	40,1 % ± 0,84 %	62,7 % ± 2,6 %	31,7 % ± 0,55 %	57,6 % ± 0,77 %	39,6 % ± 5,2 %	15,4 % ± 0,73 %
DBN	71,7 % ± 5,7 %	42,5 % ± 1,5 %	36,0 % ± 1,6 %	54,1 % ± 1,2 %	32,6 % ± 1,1 %	41,1 % ± 5,2 %	23,6 % ± 1,0 %
EBN	93,6 % ± 2,1 %	46,3 % ± 0,35 %	59,1 % ± 2,4 %	57,0 % ± 0,72 %	41,7 % ± 0,77 %	41,5 % ± 4,6 %	29,5 % ± 2,4 %
Viterbi	93,8 % ± 3,3 %	91,1 % ± 1,5 %	72,6 % ± 2,1 %	91,6 % ± 1,3 %	71,2 % ± 3,2 %	73,1 % ± 2,0 %	73,5 % ± 1,3 %

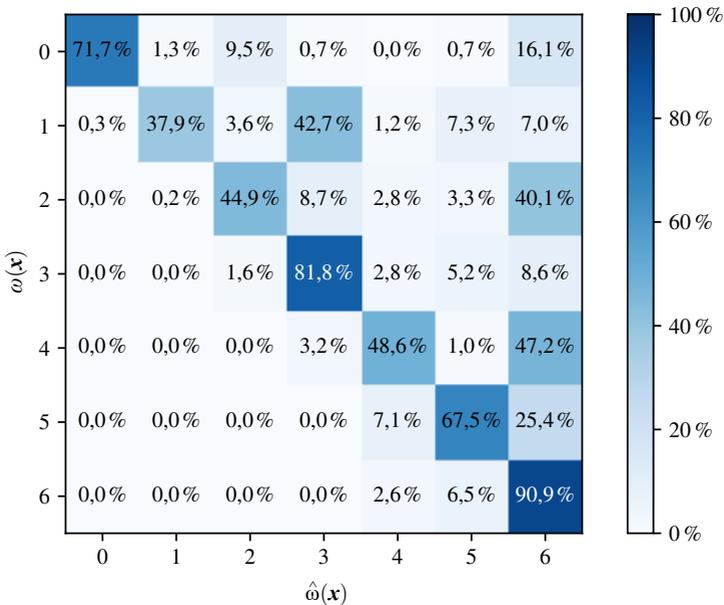


Abbildung 5.37: Normalisierte Konfusionsmatrix der Klassifikationsergebnisse des DBN. Eine Zeile zeigt die wahre Klassenzugehörigkeit eines Merkmalsvektors, während eine Spalte die geschätzte Klassenzugehörigkeit widerspiegelt.

Vergleicht man in beiden Matrizen die Spalten 1 und 5, so fallen ähnliche Verhältnisse zwischen korrekter Klassifikation und einem Verschmieren von Fehlklassifikationen im jeweils restlichen Teil der Spalte ab.

Für die Phase 1, bedeutet das beispielsweise, dass beide Modelle bei etwa 1% der Merkmalsvektoren der Phase 0 eine falsche (zu frühe) Einordnung zu Phase 1 vorgenommen wurde (vgl. Matrizen in der Zeile 0 und Spalte 1 mit Werten 1,3% bzw. 1,1%).

Gleichzeitig werden in beiden Modellen knapp 0,2% der Merkmalsvektoren der Phase 2 irrtümlich (noch) als Phase 1 klassifiziert (vgl. Matrizen in der Zeile 2 und Spalte 1 mit Werten 0,2% bzw. 0,3%).

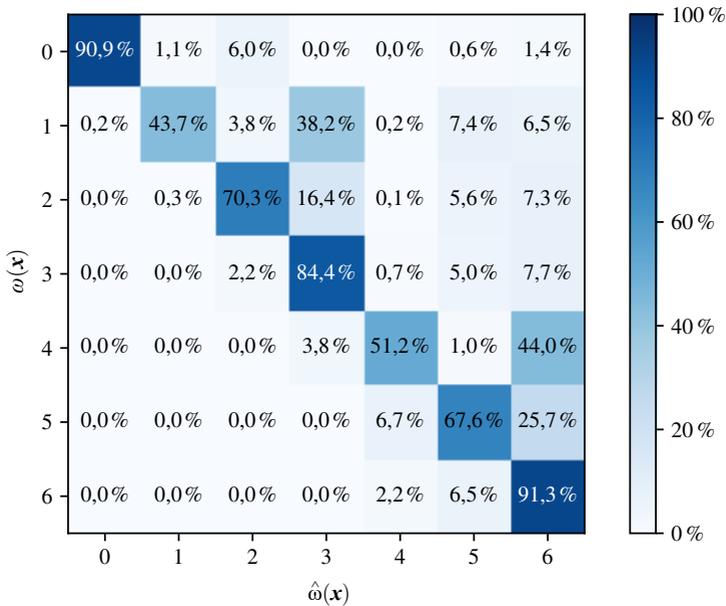


Abbildung 5.38: Normalisierte Konfusionsmatrix der Klassifikationsergebnisse des EBN. Eine Zeile zeigt die wahre Klassenzugehörigkeit eines Merkmalsvektors, während eine Spalte die geschätzte Klassenzugehörigkeit widerspiegelt.

Auch in den Zeilen 4, 5 und 6 beider Matrizen zeigen sich ähnliche Einträge. Schwächen beider Modelle liegen in der Konfusion der Merkmalsvektoren der Phase 4 mit der letzten Phase 6.

In den restlichen Bereichen der Matrizen ergeben sich teils deutliche Unterschiede. Das EBN zeigt dort weniger Konfusion an den Rändern (also weit entfernten Phasen) und zeigt stärkere Ausprägungen auf den Diagonalelementen. Die Phase 2 ist in diesem Zusammenhang besonders hervorzuheben.

In der Gesamtschau ergeben sich die Ergebnisse in Tabelle 5.15 und Abbildung 5.39 auf Seite 195. Dabei zeigt sich das EBN in Summe für alle Kennwerte signifikant besser als DBN und RF.

Tabelle 5.15: Die Tabelle zeigt in den Spalten die Gesamtergebnisse der Kreuzvalidierung bezüglich *Precision*, *Recall* und *Jaccard-Score* (Jaccard). Weiterhin sind jeweils die empirischen Standardabweichungen angegeben. Hinweis: Eine alternative Darstellung der Ergebnisse findet sich in Abbildung 5.39.

Modell	Precision	Recall	Jaccard
RF	54,4 % ± 1,0 %	57,0 % ± 1,3 %	51,8 % ± 1,1 %
DBN	61,8 % ± 0,72 %	63,4 % ± 1,1 %	60,5 % ± 0,7 %
EBN	67,8 % ± 0,43 %	71,8 % ± 0,27 %	66,9 % ± 0,43 %
Viterbi	87,1 % ± 1,1 %	90,6 % ± 1,0 %	92,4 % ± 1,2 %

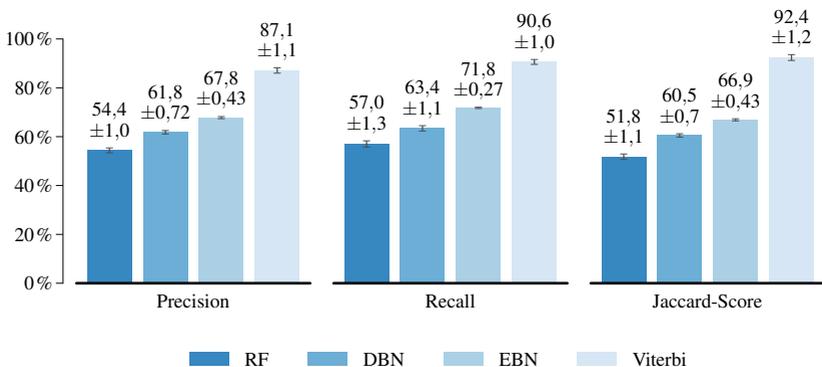


Abbildung 5.39: Die Abbildung zeigt die Gesamtergebnisse der Kreuzvalidierung bezüglich *Precision*, *Recall* und *Jaccard-Score* - also den Kennwerten zur Bewertung der Problemlösung (vgl. Kapitel 5.2.2). Darüber hinaus sind jeweils die empirischen Standardabweichungen angegeben. Dabei zeigt sich der RF als signifikant schlechter als die Vergleichsmodelle. Ebenso zeigt das DBN signifikant schlechtere Ergebnisse als das EBN.

6 Schlussbetrachtung

Die Beiträge der vorliegenden Dissertation zur Formalisierung und Analyse von medizinischen Prozessen lassen sich thematisch in den Bereichen der Unterstützung bei Diagnosestellung, sowie der Assistenz bei chirurgischen Interventionen verorten. Es ist abzusehen, dass zukünftig in diesen Bereichen viele Aspekte der klinischen Praxis und des Gesundheitswesens durch KI entscheidend mitgeprägt werden [Yu18].

Datenbasierte KI-Methoden, insbesondere Verfahren aus dem Bereich des Maschinellen Lernens, brillieren üblicherweise, wenn große und strukturierte Datenmengen verfügbar sind. Anhand großer Lern-, Validierungs- und Teststichproben können so zum Beispiel im Kontext der bildgebenden Diagnostik unterschiedlich parametrisierte Modelle eingelernt und beurteilt werden.

Betrachtet man jedoch allgemein den Bereich der Diagnostik, stehen geeignete Stichproben häufig nur eingeschränkt und/oder in unstrukturierter Form zur Verfügung. Deshalb werden Verfahren zur Strukturierung und zum Herstellen von Interoperabilität immer wichtiger.

In diesem Zuge gewann FHIR zunehmend an Bedeutung [Leh19a, Hub20]. Ziel von FHIR ist es, medizinische Daten weltweit auszutauschen, zu sammeln und mit Methoden der KI nutzen zu können [Leh19b]. Medizinische Daten können so auffindbar (engl. Findable), zugänglich (engl. Accessible), interoperabel (engl. Interoperable) sowie wiederverwendbar (engl. Reusable) genutzt werden – also sprich: nach dem sogenannten FAIR-Prinzip [Wil16, Thu21].

Abgeleitet aus den beschriebenen Entwicklungen, nutzte die vorliegende Dissertation FHIR zur Wissensrepräsentation im Rahmen der Prozessmodellierungen. Weiterhin wurde gezeigt, wie gezielt vorhandenes Expertenwissen genutzt, um auch in Bereichen, die datenarm aber wissensreich sind, performante probabilistische Abgleichmodelle bereitstellen zu können. Im medizinischen Kontext sind hier speziell die Bereiche der Diagnosestellung oder der chirurgischen Interventionen zu nennen. Es zeigt sich, dass mithilfe der vorgestellten Ansätze implizit vorhandenes Prozesswissen formalisiert und eine praxistaugliche Assistenz konzipiert und bereitgestellt werden kann.

6.1 Fazit

In dieser Arbeit wurden zunächst die Grundlagen der wissensbasierten Modellierung beleuchtet. Dabei wurde auf Expertensysteme und deren Verbindung zur KI eingegangen. Im weiteren Verlauf sind Bayes-Netze und FHIR ausführlich vorgestellt worden. Letztere wurden hinsichtlich einer Wissensrepräsentation in Zusammenhang mit einer medizinischen Entscheidungsunterstützung betrachtet. Dabei wurde ein erster Beitrag dieser Arbeit, die probabilistische Erweiterung eines FHIR DataRequirements, erläutert.

Schließlich wurde im Detail auf die Prozessmodellierung eingegangen, hierin wurden allgemeine Definition, Petri-Netze, UML-Aktivitätsdiagramme sowie Prozessbäume näher beleuchtet. Damit wurde der Grundstein zum Verständnis der nachfolgenden Kapitel der Arbeit gelegt.

Im weiteren Verlauf wurde auf den zweiten Beitrag, die Transformation zwischen Prozessbäumen und FHIR, eingegangen. Dabei konnte gezeigt werden, dass die dargelegte Transformation sowohl für die Hin- als auch für die Rücktransformation strukturerhaltend ist. Darüber hinaus bettet sich die vorgestellte Transformation in ein entworfenes Gesamtkonzept (dritter Beitrag) ein, in welchem sie als entscheidendes Puzzlestück für die Überführung von sowohl Expertenwissen als auch von Ergebnissen datenbasierter Methoden zur Entdeckung von Abläufen dient.

Ausgehend von diesen Ergebnissen wurden zwei prototypische Vertreter chirurgischer Interventionen vorgestellt: die endoprothetische Versorgung eines Patienten mittels einer totalen Endoprothese der Hüfte, sowie die laparoskopische Cholezystektomie, ein chirurgisches Verfahren zur Entfernung (Ektomie) der Gallenblase (engl. Cholecyst). Ausgehend vom Stand der Technik zur medizinischen Phasendetektion wurde die entwickelte Transformation eines FHIR-Plans in ein expertenbasiertes DBN vorgestellt (vierter Beitrag). Das Modell wurde in einer Vor- und Realstudie sowie an einem öffentlichen Datensatz einer Leistungsbewertung unterzogen. Dabei stellte sich das expertenbasierte DBN mit solider Leistung unter Berücksichtigung der gewählten Leistungswerte, Precision, Recall und Jaccard-Score heraus.

6.2 Ausblick

Die in dieser Arbeit vorgestellte Transformation zur medizinischen Phasendetektion kann in Anlehnung an die bereits getätigten eigenen Veröffentlichungen erweitert werden, vgl. [Phi15a, Phi15b, Phi15c, Phi16a, Phi16b, Phi16c, Sch16, Phi17a, Phi17b, Phi17c, Phi18a, Phi18b, Phi18c, Phi18e, Phi18d, Phi19a, Phi19b, Phi19c, Phi20, Bom21b, Phi21, Bom21a, Phi22]. Interessant ist dies vor allem im Bereich der Diagnose, die nicht nur einen einzigen, isolierten Mediziner betrifft, sondern ein Zusammenspiel mehrerer Disziplinen beinhaltet.

Durch die steigende Komplexität in der medizinischen Diagnostik und Therapie sind künftig immer mehr Szenarien denkbar, in denen Medizinerinnen und Mediziner verschiedener Disziplinen gemeinsam an einem klinischen Fall arbeiten. Allerdings nicht mehr isoliert voneinander mit Synchronisationen nur in längeren Abständen, sondern mit der Möglichkeit, jederzeit bei der eigenen Arbeit auf die gegenwärtig vorhandenen Patientendaten, sowie auf die Ergebnisse der Kollegen und eines Entscheidungsunterstützungssystems zuzugreifen. So könnte beispielsweise ein Radiologe vor dem Verfassen seines Berichtes bereits auf die Ergebnisse und die abgeleitete Handlungsempfehlung der nur Stunden zuvor erstellten Gewebeauswertung der betreffenden Patientin zugreifen und so seine Tätigkeiten sofort in die richtige Richtung lenken. In der derzeit herrschenden Arbeitsweise erfahren Ärztinnen und Ärzte in der

Regel erst im Nachhinein, z.B. im Rahmen eines wöchentlichen Tumorboards, von aktuellen Entwicklungen. Durch die beschriebenen neuen, kooperativen Ansätze ist zu erwarten, dass die Diagnose- und Therapiequalität verbessert und die zeitlichen Abläufe stark verkürzt werden können.

Derlei Ansätze sind für Praxisnetzwerke, ländliche Praxen, aber auch für große Klinikverbünde von großem Interesse, wie die Konsortien der eigenen Projekte und Projektanträge zeigen. Hier können die vorgestellten Beiträge wichtige Grundpfeiler sein, auf deren Erweiterungen maßgeschneiderte, multidisziplinäre Assistenzfunktionen bereitgestellt werden können.

Zusätzlich ergeben sich in der medizinischen Praxis bei der Formalisierung der Daten neue Herausforderungen. Betrachtet man das Beispiel der Pathologie, so werden hier aktuell im Rahmen von FHIR (und daran angeschlossen auch SNOMED CT) neue, noch nicht vorhandene Datenstrukturen geschaffen, um den in dieser Disziplin eigenen Prozessen und Artefakten gerecht zu werden. Während dieser Bestrebungen entstehen durch den technischen Fortschritt in diesem medizinischen Bereich fortwährend neue Ideen für zusätzliche Strukturen. Beispielsweise erzielt die sektorenweise, automatische Auswertung von Tumorgewebe deutlich genauere Ergebnisse als dies bislang durch die Kennzeichnung des Gewebes als insgesamt tumorös üblich war. Durch den erhöhten Detaillierungsgrad erfordert diese Technik jedoch auch neue Strukturen zur Formalisierung. Dies führt in den entsprechenden Gremien unweigerlich zur Bewertung des zusätzlichen Aufwands im Vergleich zum Nutzen: „Welche Vorteile bringt die Erfassung und Formalisierung dem Medizinischen Fachpersonal?“ Also sprich: „Wo bleibt die Verknüpfung mit Entscheidungsunterstützungssystemen und wie hoch ist der Benefit (Quickwin oder langfristige Vorteile) von derlei Aktivitäten?“

Um diese dringenden Fragen zu adressieren, bedarf es abseits der Hinterlegung bloßer (Untersuchungs-)Werte auch einer Formalisierung und Hinterlegung des medizinischen Prozesswissens, das die einzelnen Untersuchungsschritte zu einem Gesamtablauf verbindet. Dieses Wissen existiert zwar implizit in den Köpfen der beteiligten Disziplinen, die Prozesse sind aber typischerweise nicht formal hinterlegt, prüfbar, austauschbar und vor allem nicht direkt verknüpfbar.

Auch hier können die in dieser Arbeit getätigten Beiträge (und deren Erweiterung) helfen und durch die Formalisierung, Hinterlegung und Verknüpfung von Expertenwissen zu medizinischen Prozessen neue Impulse setzen, wie auch das Engagement auf Konferenzen, der DIN und SNOMED International zeigen.

Weiterhin müssen Entscheidungen immer noch von menschlichen Experten getroffen werden, auch wenn sie von KI-basierten Systemen unterstützt werden. Der effiziente und effektive Einsatz solcher Assistenzsysteme setzt voraus, dass Domänenexperten verstehen, wie eine KI-Lösung zu einer Empfehlung, z.B. einer bestimmten Therapie, kommt. Mit zunehmender Schwere der Entscheidungen (z.B. im Rahmen der Diagnose und Therapie onkologischer Erkrankungen) kann das Vertrauen in KI-Systeme schnell sinken. Besonders vielversprechend ist jedoch gerade der Einsatz von Methoden der KI in der Krebsbehandlung, da Tumorerkrankungen in Zukunft in vielen Fällen nicht mehr nach ihrer Lokalisation, sondern nach ihrem genomischen Profil behandelt werden. Diagnostische und therapeutische Entscheidungen werden also zu einem großen Teil auf der molekulargenetischen Tumoranalyse von Patienten beruhen, die durch KI ideal unterstützt werden kann. Hierzu ist es allerdings notwendig, Faktoren der Akzeptierbarkeit¹ zu betrachten, die für Entscheidungsträger (z.B. Ärzte, Patienten, ...) bei genomisch-onkologischen Diagnose- und Therapieentscheidungen besonders relevant sind. KI-basierte Fehlentscheidungen im medizinischen Kontext liegen in einem besonders sensiblen Bereich: Beispielsweise kann eine von einem KI-System empfohlene Röntgenuntersuchung bei einer schwangeren Frau entgegen medizinischer Prozesse geltender Leitlinien stehen. Im Sinne der Kontrollierbarkeit können medizinische Prozesse, welche mit der in dieser Arbeit vorgestellten Vorgehensweise formalisiert wurden, als Leitplanken für KI-Algorithmen dienen und deren Ergebnismenge einschränken.

¹ Hinweis: Akzeptanz (Acceptance) entsteht beim Menschen selbst. Um sie günstig zu beeinflussen, sollte nach technischen Ansätzen mit einem hohen Grad der Akzeptierbarkeit (Acceptability) gestrebt werden. Möglichkeiten dies technisch umzusetzen, bestehen beispielsweise in den Bereichen der Erklärbarkeit, Transparenz, Fairness und Kontrollierbarkeit.

Abkürzungen

AI	Artificial Intelligence S. 13, 14
ANSI	American National Standards Institute S. 34
BN	Bayes-Netz S. 25, 30, 31
BPM	Geschäftsprozessmodellierung (engl. Business Process Modeling) S. 51
BPMN	Geschäftsprozessmodell und -notation (engl. Business Process Model and Notation) S. 52, 82, 83
CAS	Computerassistierte Chirurgie (engl. Computer Assisted Surgery) S. 101
CNN	Faltendes Neuronales Netz (engl. Convolutional Neural Network) S. 149
CPD	Conditional Probability Distribution – bedingte Wahrscheinlichkeitsverteilung S. 26
CPT	Conditional Probability Table – bedingte Wahrscheinlichkeitstabelle S. 26
DAG	gerichteter (engl. directed) azyklischer Graph S. 25–28, 30, 74

DBN	Dynamisches Bayes-Netz S. ii, 31–33, 143–145, 147, 149, 150, 152–156, 164, 165, 167, 168, 178, 180, 181, 183–185, 189, 191, 193–195, 199
DIN	Deutsches Institut für Normung e.V. S. 201
EBN	Expertenbasiertes DBN S. 178, 180, 181, 183–186, 189, 191, 194, 195
ePA	Elektronische Patientenakte S. i, 2, 3, 35, 37
EPK	Ereignisgesteuerte Prozesskette S. 52, 82
FHIR	Fast Healthcare Interoperability Resources S. i, ii, 2–5, 34–39, 69, 73, 81–83, 88–90, 92, 150, 197–200
GOF AI	Good Old-Fashioned Artificial Intelligence S. 14
HL7	Health Level Seven S. 34
HMM	Hidden Markov Modell S. 115, 142, 143, 147, 149, 154
ISO	International Organization for Standardization S. 34
KI	Künstliche Intelligenz S. 1–3, 7, 11, 13–16, 35, 197, 198, 201
KNN	Künstliche Neuronale Netze S. 149, 150
LOINC	Logical Observation Identifiers Names and Codes S. 38, 43
LSTM	Langes Kurzzeitgedächtnis (engl. Long Short-Term Memory) S. 149
MIO	Medizinisches Informationsobjekt S. 3, 37

OMG	Object Management Group S. 63
OSI	Open Systems Interconnection S. 34
RF	Random Forest S. 132–137, 142, 146, 147, 164, 166, 168, 178, 180–184, 189, 194, 195
RNN	Rekurrentes Neuronales Netz S. 149
ROS	Robot Operating System S. 163, 170
SMART	Substitutable Medical Applications Reusable Technologies S. 35
SNOMED	Systematized Nomenclature of Medicine S. 38, 43, 200, 201
SPN	Stochastisches Petri-Netz S. 139–141, 147
TBN	Temporal Bayesian Network S. 32
TEP	Totale Endoprothese S. 103–105, 107, 108, 113, 114, 135, 163, 164, 168–170, 208
UML	Unified Modeling Language S. 63, 73, 82, 83, 114, 173
WBS	Wissensbasiertes System S. 7–9, 81

Medizinischer Glossar

- Calot'sches Dreieck** Bereich zwischen Leber, Gallenwege und Arterien. Dient als Orientierungshilfe bei der Cholezystektomie. | S. 110, 111
- Cholezystektomie** Chirurgisches Verfahren zur Entfernung (Ektomie) der Gallenblase (engl. Cholecyst). Nicht zu verwechseln mit einer Cholezystostomie, also dem Öffnen (Stomie) der Gallenblase. | S. 108–112, 134, 143, 149, 186, 187, 189, 199, 207, 208
- Elektro-kauter** Chirurgisches Instrument, welches im Rahmen einer Elektrokoagulation eingesetzt werden kann um Gewebe zu durchtrennen. | S. 171, 178, 179
- Elektrokoagulation** Zerstörung von Gewebe mittels Strom. Wird häufig zur Blutstillung und/oder zur Durchtrennung von Gewebe eingesetzt. | S. 111, 171, 207
- Kaudal** Zum Steißbein hin orientiert. | S. 165, 172
- Kranial** Zum Schädel hin orientiert. | S. 104, 165, 172
- Laparoskop** Ein spezielles Endoskop, welches für die Durchführung einer Laparoskopie, also einer Bauchspiegelung, verwendet wird. | S. 109, 186

- Pneumo-peritoneum** Ansammlung von Gasen in der Bauchhöhle. Eine Ansammlung von Kohlendioxid oder Helium wird beispielsweise bei einer Insufflation der Bauchhöhle im Rahmen einer laparoskopischen Cholezystektomie künstlich herbeigeführt. | S. 109, 110
- Totale Endoprothese** Die endoprothetische Versorgung eines Patienten mittels einer totalen Endoprothese (siehe TEP) , ist ein Eingriff, bei dem Körperteile vollständig (=total) durch eine Prothese (=künstliches Körperteil) ersetzt werden. Diese Prothese verbleibt im Inneren (=endo) des Menschen als sogenannte Endoprothese. Häufig werden Hüft- oder Kniegelenke endoprothetisch versorgt – man spricht dann von Hüft- bzw. Knie-TEPs. | S. 169, 205
- Trokar** Spitzer Dorn mit umgebender Hülse. Dient zur Schaffung künstlicher Körperöffnungen. | S. 43, 44, 110
- Viererposition** Position eines Patienten im Rahmen einer Hüft-TEP. Der Unterschenkel des betroffenen Beines wird dabei im rechten Winkel auf dem Oberschenkel des anderen Beines positioniert, sodass der Eindruck einer Vier entsteht. | S. 106, 108

Symbolverzeichnis

Allgemein

$\mathbf{x}, \mathbf{y}, \mathbf{z}$	Vektoren	klein, kursiv, fett
$\mathbf{X}, \mathbf{Y}, \mathbf{Z}$	Matrizen	groß, kursiv, fett
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}$	Mengen	groß, kalligraphisch
$\underline{X}, \underline{Y}, \underline{Z}$	Zufallsvariablen	groß, kursiv, unterstrichen
X, Y, Z, x, y, z	Bezeichner	groß oder klein, kursiv
$\mathbf{x}_n, \mathbf{X}_n, \underline{X}_n, X_n$	Vektor, Matrix, Zufallsvariable und Bezeichner mit Index n	
$\mathbf{x}^t, \underline{X}^t, X^t$	Vektor, Zufallsvariable und Bezeichner zum Zeitschritt t	
$\mathbf{x}_n^t, \underline{X}_n^t, X_n^t$	Vektor, Zufallsvariable und Bezeichner mit Index n zum Zeitschritt t	
$\mathbf{x}_{0:n}, \underline{X}_{0:n}$	Vektoren und Zufallsvariablen gebündelt in einer Sequenz $\mathbf{x}_0, \dots, \mathbf{x}_n$ bzw. $\underline{X}_0, \dots, \underline{X}_n$ von Index 0 bis Index n	
$\mathbf{x}^{0:t}, \underline{X}^{0:t}$	Vektoren und Zufallsvariablen gebündelt in einer Sequenz $\mathbf{x}^0, \dots, \mathbf{x}^t$ bzw. $\underline{X}^0, \dots, \underline{X}^t$ von Zeitschritt 0 bis Zeitschritt t	
$\mathbf{x}^\top, \mathbf{X}^\top$	Vektor und Matrix transponiert	

Speziell

\emptyset	Leere Menge
α, β, γ	Ereignisse
A	Übergangsmatrix
$a_{i,j}$	Eintrag der Matrix A in Zeile mit Index i und Spalte mit Index j
\mathcal{A}	Menge der Aktionen eines Prozesses
B	Matrix der Emissionswahrscheinlichkeiten
$b_{i,j}$	Eintrag der Matrix B in Zeile mit Index i und Spalte mit Index j
\mathcal{B}_{Inp}	Menge der Eingabe-Datenobjekte eines Prozessbaums
\mathcal{B}_{Out}	Menge der Ausgabe-Datenobjekte eines Prozessbaums
\mathcal{B}	Menge der Entscheidungs- und Verbindungsknoten eines UML Aktivitätsdiagramms
\mathcal{B}_{Ent}	Menge der Entscheidungsknoten eines UML Aktivitätsdiagramms
\mathcal{B}_{Ver}	Menge der Verbindungsknoten eines UML Aktivitätsdiagramms
C	Konfusionsmatrix
$c_{i,j}$	Eintrag der Konfusionsmatrix C in Zeile mit Index i und Spalte mit Index j
\mathcal{C}	Menge der Gabelungs- und Synchronisationsknoten eines UML Aktivitätsdiagramms
\mathcal{C}_{Gab}	Menge der Gabelungsknoten eines UML Aktivitätsdiagramms

\mathcal{C}_{Syn}	Menge der Synchronisationsknoten eines UML Aktivitätsdiagramms
$\text{Chi}(v)$	Menge der Kindknoten eines Knotens v
$\mathcal{D}_{\text{Lern}}$	Lernstichprobe
$\mathcal{D}_{\text{Test}}$	Teststichprobe
\mathcal{D}_{Val}	Validierungsstichprobe
\mathcal{D}_i	i -te Stichprobe
$\text{Deg}^+(v)$	Ausgangsgrad eines Knotens v
$\text{Deg}^-(v)$	Eingangsgrad eines Knotens v
$\text{Desc}(v)$	Menge der Nachfahren eines Knotens v
ESN	Elementares Systemnetz
\mathcal{E}	Kantenmenge des Graphen eines BN
$\mathcal{E}_{B_0}, \mathcal{E}_{B_{\rightarrow}}$	Kantenmengen des Graphen eines DBN
\mathcal{F}	Flussrelation eines Petri-Netzes
$G = (\mathcal{V}, \mathcal{E})$	Graph eines BN mit Knotenmenge \mathcal{V} und Kantenmenge \mathcal{E}
$\mathcal{G}_{\rightarrow}$	Menge der gruppierenden Aktionen für Sequenzen eines FHIR-Plans
$H = (\mathcal{K}, \mathcal{M})$	Graph eines FHIR-Plans mit Knotenmenge \mathcal{K} und Kantenmenge \mathcal{M}
\mathcal{H}	Menge möglicher Emissionen eines HMM
i, j, k, l, m, n	Ganzzahlige Indizes
\mathcal{J}	Menge des Startknotens eines UML Aktivitätsdiagramms
\mathcal{J}	Menge der elementaren Aktionen eines FHIR Plans
\mathcal{J}_{τ}	Menge der stillen Aktionen eines FHIR Plans
$\mathcal{J}_{\text{Start}}$	Menge der Startaktion eines FHIR Plans

$\mathcal{J}_{\text{Ende}}$	Menge der Endaktion eines FHIR Plans
\mathcal{K}	Knotenmenge eines FHIR Plans
\mathcal{L}	Menge der Schaltraten
λ_i	Schaltrate mit Index i
$\mathcal{M}_0(\cdot)$	Anfangsmarkierung, Markierung mit Index 0
$\mathcal{M}(\cdot)$	Markierung
\mathcal{M}	Kantenmenge eines FHIR Plans
\mathcal{M}_{In}	Menge der Kanten eines FHIR Plans, die Aktionen mit einem ihnen zugeordneten Dateninput verbinden
$\mathcal{M}_{\text{Nach}}$	Menge der Kanten eines FHIR Plans, die eine sequentielle Verschaltung von Aktionen repräsentieren
\mathcal{M}_{Out}	Menge der Kanten eines FHIR Plans, die Aktionen mit einem ihnen zugeordneten Datenoutput verbinden
\mathcal{M}_{Vor}	Menge der Kanten eines FHIR Plans, die repräsentieren, dass eine Aktion vor dem Beginn einer anderen abgeschlossen sein muss
\mathcal{M}_{Sub}	Menge der Kanten eines FHIR Plans, die gruppierende Aktionen mit ihren Subaktionen verbinden
\mathbb{N}	Menge der natürlichen Zahlen einschließlich Null
\mathcal{N}	Knotenmenge eines UML Aktivitätsdiagramms
NST	Netzstruktur
\mathcal{O}	Menge der Objektknoten eines UML Aktivitätsdiagramms
\mathcal{O}_{Imp}	Menge der Dateninputs eines FHIR Plans
\mathcal{O}_{Out}	Menge der Datenoutputs eines FHIR Plans
ω	Klasse oder Ergebnis

$\omega(\mathbf{x}), y$	Wahre Klassenzugehörigkeit des Merkmalsvektors \mathbf{x} , oder auch: Label
$\hat{\omega}(\mathbf{x}), \hat{y}$	Geschätzte Klassenzugehörigkeit des Merkmalsvektors \mathbf{x} , oder auch: Klassifikation
Ω	Klassen- oder Ergebnisraum
\mathcal{P}	Menge der Plätze
$P(\cdot)$	Wahrscheinlichkeit
$P(\cdot, \cdot)$	Verbundwahrscheinlichkeit
$P(\cdot \cdot)$	Bedingte Wahrscheinlichkeit
$Pa(v)$	Menge der Elternknoten eines Knoten v
p	Platz
π	Vektor der Startverteilung
\mathcal{Q}	Menge der Transitionen
q	Transition
$\cdot^* q$	Vorbereich der Transition q
q^*	Nachbereich der Transition q
\mathcal{R}	Kantenmenge eines UML Aktivitätsdiagramms
\mathcal{R}_{Dat}	Datenfluss eines UML Aktivitätsdiagramms
\mathcal{R}_{Kon}	Kontrollfluss eines UML Aktivitätsdiagramms
$\rho(u)$	Reihenfolge der Kindknoten des Knotens u
\mathcal{S}	Menge verborgener Zustände eines HMM
t	Zeitschritt
$T = (\mathcal{X}, \mathcal{Y})$	Graph eines Prozessbaums mit Knotenmenge \mathcal{X} und Kantenmenge \mathcal{Y}
τ	Stille Aktion eines Prozessbaumes
\mathcal{T}	Menge der stillen Aktionen eines Prozessbaumes

$U = (\mathcal{N}, \mathcal{R})$	Graph eines UML Aktivitätsdiagramms mit Knotenmenge \mathcal{N} und Kantenmenge \mathcal{R}
\mathcal{V}	Knotenmenge des Graphen eines BN
$\mathcal{V}_{B_0}, \mathcal{V}_{B_{\rightarrow}}$	Knotenmengen des Graphen eines DBN
$Val(\underline{X})$	Wertebereich einer Zufallsvariable \underline{X}
$ Val(\underline{X}) $	Anzahl der Werte des Wertebereichs einer Zufallsvariablen \underline{X}
$w_G^{u \rightarrow v}$	Weg innerhalb des gerichteten Graphen G vom Knoten u bis zum Knoten v gegeben als Folge von Knoten
$\mathcal{W}_G^{u \rightarrow v}$	Knotenmenge auf dem Weg vom Knoten u zum Knoten v innerhalb des gerichteten Graphen G ohne den Anfangsknoten u
\mathbf{x}	Merkmalsvektor
\mathcal{X}	Knotenmenge eines Prozessbaums
\mathcal{Y}	Kantenmenge eines Prozessbaums
\mathcal{Z}	Menge des Endeknotens eines UML Aktivitätsdiagramms
\oplus	Menge möglicher Operatortypen eines Prozessbaumes
\oplus°	Menge der Operatorknoten eines Prozessbaumes
\wedge	Nebenläufiger Verschaltungsoperator eines Prozessbaumes
\times	Selektiver Verschaltungsoperator eines Prozessbaumes
\rightarrow	Sequentieller Verschaltungsoperator eines Prozessbaumes
\circlearrowleft	Schleifen-Operator eines Prozessbaumes

Literatur

- [Aal05] AALST, Wil van der and LASSEN, Kristian Bisgaard: „Translating Workflow Nets to BPEL“. In: *BETA Working Paper Series, WP 145* (2005).
- [Aal11] AALST, Wil van der; BUIJS, Joos and DONGEN, Boudewijn van: „Towards Improving the Representational Bias of Process Mining“. In: *International Symposium on Data-Driven Process Discovery and Analysis*. Berlin: Springer, 2011, S. 39–54.
- [Aal13] AALST, Wil van der: „Business Process Management: A comprehensive Survey“. In: *International Scholarly Research Notices* 30 (2013), S. 1–37.
- [Aal16] AALST, Wil van der: *Process Mining: Data Science in Action*. Berlin: Springer, 2016.
- [Aal96] AALST, Wil van der: „Structural Characterizations of Sound Workflow Nets“. In: *Computing Science Reports* 96.23 (1996), S. 18–22.
- [Aal97] AALST, Wil van der: „Verification of Workflow Nets“. In: *International Conference on Application and Theory of Petri Nets*. Heidelberg, 1997, S. 407–426.
- [Aal98] AALST, Wil van der: „The Application of Petri Nets to Workflow Management“. In: *Journal of Circuits, Systems, and Computers* 8.01 (1998), S. 21–66.
- [Ake09] AKERKAR, Rajendra and SAJJA, Priti: *Knowledge-based Systems*. Boston: Jones & Bartlett Publishers, 2009.

- [Alt13] ALTENKRÜGER, Doris and BÜTTNER, Winfried: Wissensbasierte Systeme: Architektur, Entwicklung, Echtzeitanwendungen-Eine praxisgerechte Einführung. Heidelberg: Springer, 2013.
- [App21] APPLE INC.: Healthcare and Health Records. Online Ressource, Zugriff: 01.08.2021. 2021. URL: <https://www.apple.com/healthcare/health-records/>.
- [Bal10] BALZERT, Helmut: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Heidelberg: Springer, 2010.
- [Bau02] BAUSE, Falko u. a.: Stochastic Petri Nets – An Introduction to the Theory. Princeton: Citeseer, 2002.
- [Bau14] BAUMGARTEN, Bernd: Kompendium der diskreten Mathematik. München: Walter de Gruyter, 2014.
- [Bau79] BAUER, Ruediger; KERSCHBAUMER, Franz; POISEL, Stefan and OBERTHALER, Werner: „The Transgluteal Approach to the Hip Joint“. In: *Archives of Orthopaedic and Traumatic Surgery* 95.1-2 (1979), S. 47–49.
- [Bei19] BEIERLE, Christoph and KERN-ISBERNER, Gabriele: Methoden wissensbasierter Systeme. Berlin: Springer, 2019.
- [Ben21] BENSON, Tim and GRIEVE, Grahame: Principles of Health Interoperability: FHIR, HL7 and SNOMED CT. Bern: Springer Nature Switzerland, 2021.
- [Bey16] BEYL, Tim; SCHREITER, Luzie; NICOLAI, Philip; RACZKOWSKY, Jörg and WÖRN, Heinz: „3D Perception Technologies for Surgical Operating Theatres“. In: *Studies in Health Technology and Informatics* 220 (2016), S. 45–50.
- [Bie13] BIETHAHN, Jörg: Entwicklung von Expertensystemen: Eine Einführung. Berlin: Springer, 2013.
- [Bih16] BIHLMAIER, Andreas; BEYL, Tim; NICOLAI, Philip; KUNZE, Mirko; MINTENBECK, Julien; SCHREITER, Luzie; BRENNECKE, Thorsten; HUTZL, Jessica; RACZKOWSKY, Jörg and WÖRN, Heinz: „Ros-based cognitive surgical robotics“. In: *Robot Operating System (ROS)*. Berlin: Springer, 2016, S. 317–342.

- [Blu08] BLUM, Tobias; PADOY, Nicolas; FEUßNER, Hubertus and NAVAB, Nassir: „Modeling and Online Recognition of Surgical Phases Using Hidden Markov Models“. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. Heidelberg: Springer, 2008, S. 627–635.
- [Bom21a] BOMMERSHEIM, Marie and PHILIPP, Patrick: „Supporting the Incorporation of Individual Patient Preferences for Decision Support in Breast Cancer Therapy“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 15-17, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2021, S. 523–525.
- [Bom21b] BOMMERSHEIM, Marie; PHILIPP, Patrick; BEYERER, Jürgen and HEMPEL, Dirk: „Workflow Analysis for Supporting Breast Cancer Therapy Using Process Mining“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, July 26-29, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2021, S. 523–528.
- [Bre01] BREIMAN, Leo: „Random Forests“. In: *International Journal of Machine Learning* 45.1 (2001), S. 5–32.
- [Bre96] BREIMAN, Leo: „Bagging Predictors“. In: *International Journal of Machine Learning* 24.2 (1996), S. 123–140.
- [Büh02] BÜHLMANN, Peter: „Bootstraps for Time Series“. In: *Statistical Science* 17.1 (2002), S. 52–72.
- [Bui12] BUIJS, Joos CAM; DONGEN, Boudewijn F van and AALST, Wil van der: „A Genetic Algorithm for Discovering Process Trees“. In: *2012 IEEE Congress on Evolutionary Computation*. IEEE. 2012, S. 1–8.
- [Bun17] BUNDY, Alan: „Preparing for the Future of Artificial Intelligence“. In: *AI & Society* 32 (2017), S. 285–287.

- [Car86] CARLSTEIN, Edward: „The Use of Subseries Values for Estimating the Variance of a General Statistic from a Stationary Sequence“. In: *The Annals of Statistics* (1986), S. 1171–1179.
- [Dar09] DARWICHE, Adnan: *Modeling and Reasoning with Bayesian Networks*. Cambridge: Cambridge University Press, 2009.
- [Dei20] DEISENROTH, Marc Peter; FAISAL, Aldo and ONG, Cheng Soon: *Mathematics for Machine Learning*. Cambridge: Cambridge University Press, 2020.
- [Del19] DELLERMANN, Dominik; EBEL, Philipp; SÖLNER, Matthias and LEIMEISTER, Jan Marco: „Hybrid Intelligence“. In: *Business & Information Systems Engineering* 61.5 (2019), S. 637–643.
- [Die10] DIESTEL, R: *Graphentheorie*. 4. Aufl. Berlin: Springer, 2010.
- [DIN16] DIN EN ISO 6385: *Grundsätze der Ergonomie für die Gestaltung von Arbeitssystemen*. Berlin: Beuth Verlag GmbH, 2016.
- [DIN99] DIN EN ISO 9241-11: *Anforderungen an die Gebrauchstauglichkeit - Leitsätze*. Berlin: Beuth Verlag GmbH, 1999.
- [Dum13] DUMAS, Marlon; LA ROSA, Marcello; MENDLING, Jan and REIJERS, Hajo: *Fundamentals of Business Process Management*. Berlin: Springer, 2013.
- [Efr79] EFRON, Bradley: „Bootstrap Methods: Another Look at the Jackknife“. In: *The Annals of Statistics* 7.1 (1979), S. 1–26.
- [Eis13] EISENFÜHR, Franz and WEBER, Martin: *Rationales Entscheiden*. Berlin: Springer, 2013.
- [Esp17] ESPARZA, Javier: „Advances in Quantitative Analysis of Free-Choice Workflow Petri Nets (Invited Talk)“. In: *24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*. Hrsg. von SCHEWE, Sven; SCHNEIDER, Thomas and WIJSEN, Jef. Bd. 90. *Leibniz International Proceedings in Informatics (LIPIcs)*. Dagstuhl: Verlag Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017, 2:1–2:6.
- [Eva19] EVA, Benjamin: „Principles of Indifference“. In: *The Journal of Philosophy* 116.7 (2019), S. 390–411.

- [Fan15] FANKHAUSER, Péter; BLOESCH, Michael; RODRIGUEZ, Diego; KAESTNER, Ralf; HUTTER, Marco and SIEGWART, Roland: „Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling“. In: *2015 International Conference on Advanced Robotics (ICAR)*. IEEE. 2015, S. 388–394.
- [Fav15] FAVRE, Cédric; FAHLAND, Dirk and VÖLZER, Hagen: „The Relationship between Workflow Graphs and Free-choice Workflow Nets“. In: *Information Systems* 47 (2015), S. 197–219.
- [Feu91] FEUSSNER, Hubertus; UNGEHEUER, Arthur; LEHR, Lars and SIEWERT, Jan: „Technik der Laparoskopischen Cholezystektomie“. In: *Langenbecks Archiv für Chirurgie* 376.6 (1991), S. 367–374.
- [FHI21] FHIR FOUNDATION: FHIR Specification v4.0.1 R4. Online Ressource, Zugriff: 01.08.2021. 2021. URL: <http://hl7.org/fhir/R4/>.
- [Fie92] FIELD, Marilyn and LOHR, Kathleen: *Guidelines for Clinical Practice: From Development to Use*. Washington: National Academies Press, 1992.
- [Fis18] FISCHER, Thomas; SCHÖLER, Thorsten; MAIBAUM, Johannes and HÖLTGEN, Stefan: *Informatik, Programmieren, Kybernetik*. Berlin: De Gruyter, 2018.
- [Fre06] FREIBURG, Universitätsklinikum: *Patienteninformation 0906 D-OT06030 – Das künstliche Hüftgelenk*. Tuttlingen: Aesculap, 2006, S. 1–20.
- [Fri01] FRIEDMAN, Jerome; HASTIE, Trevor and TIBSHIRANI, Robert: *The Elements of Statistical Learning*. Bd. 1. 10. New York: Springer, 2001.
- [Fri06] FRIGG, Roman and HARTMANN, Silvia: „Models in Science“. In: *Stanford Encyclopedia of Philosophy* (2006).
- [Gei90] GEIGER, Dan; VERMA, Thomas and PEARL, Judea: „Identifying Independence in Bayesian Networks“. In: *Networks* 20.5 (1990), S. 507–534.
- [Goo16] GOODFELLOW, Ian; BENGIO, Yoshua and COURVILLE, Aaron: *Deep learning*. Cambridge: MIT press, 2016.

- [Goo21] GOOGLE CLOUD SERVICES: Cloud Healthcare API. Online Ressource, Zugriff: 01.08.2021. 2021. URL: cloud.google.com/healthcare/.
- [Har19] HAROSKE, Gunter and MÖRZ, Michael: Leitfaden Pathologie - Workflow in der Digitalen Medizin, Version 1.0. Hamburg: Bundesverband Deutscher Pathologen e.V., Juni 2019, S. 1–28.
- [Hau89] HAUGELAND, John: Artificial Intelligence: The Very Idea. Boston: MIT Press, 1989.
- [Her03] HERFARTH, Christian: „Lean Surgery through Changes in Surgical Workflow“. In: *British Journal of Surgery* 90.5 (2003), S. 513–514.
- [Hes84] HESSE, Wolfgang; KEUTGEN, Hans; LUFT, Alfred and ROMBACH, Dieter: „Ein Begriffsystem für die Softwaretechnik. Vorschlag zur Terminologie“. In: *Informatik Spektrum: Vol. 7, No. 4* (1984).
- [Hoc10] HOCHSTÄTTLER, Winfried: Algorithmische Mathematik. Berlin: Springer, 2010.
- [Hop92] HOPPE, Uwe: „Wissensbasierte Systeme, Wissensakquisition und Knowledge Engineering“. In: *Methoden des Knowledge Engineering*. Berlin: Springer, 1992, S. 6–31.
- [Hub20] HUBER, Lukas; HONEDER, Thomas and HACKL, Werner: „FHIR Analytics-Pragmatic Review of Recent Studies“. In: *Biomedical Informatics for Health and Care* (2020), S. 110–112.
- [Inf18] INFORMATION TECHNOLOGY INDUSTRY COUNCIL (ITI): Cloud Healthcare Pledge. Online Ressource, Zugriff: 01.08.2021. 2018. URL: <https://www.itic.org/public-policy/CloudHealthcarePledge.pdf>.
- [ISO16] ISO 6385: Ergonomics Principles in the Design of Work Systems. Genf: International Organization for Standardization Publishing, 2016.
- [Jac12] JACCARD, Paul: „The Distribution of the Flora in the Alpine Zone“. In: *New Phytologist* 11.2 (1912), S. 37–50.
- [Jac95] JACOBS, Robert A: „Methods for Combining Experts’ Probability Assessments“. In: *Neural computation* 7.5 (1995), S. 867–888.

- [Jay88] JAYNES, Edwin Thompson: „The Relation of Bayesian and Maximum Entropy mMethods“. In: *Maximum-entropy and Bayesian Methods in Science and Engineering*. Berlin: Springer, 1988, S. 25–29.
- [Jef46] JEFFREYS, Harold: „An Invariant Form for the Prior Probability in Estimation Problems“. In: *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 186.1007 (1946), S. 453–461.
- [Joy83] JOYCE, Joseph and LAPINSKY, George: „A History and Overview of the Safety Parameter Display System Concept“. In: *IEEE Transactions on Nuclear Science* 30.1 (1983), S. 744–749.
- [Kam16] KAMAR, Ece: „Directions in Hybrid Intelligence: Complementing AI Systems with Human Intelligence.“ In: *Proceedings of IJCAI*. 2016, S. 4070–4073.
- [Kas21] KASSENÄRZTLICHE BUNDESVEREINIGUNG DEZERNAT DIGITALISIERUNG UND IT: KBV-Basis-Profil. Online Ressource, Zugriff: 01.08.2021. 2021. URL: <https://mio.kbv.de/display/BASE1X0>.
- [Kat16] KATIĆ, Darko; SCHUCK, Jürgen; WEKERLE, Anna-Laura; KENNGOTT, Hannes; MÜLLER-STICH, Beat Peter; DILLMANN, Rüdiger and SPEIDEL, Stefanie: „Bridging the Gap Between Formal and Experience-Based Knowledge for Context-Aware Laparoscopy“. In: *International Journal of Computer Assisted Radiology and Surgery* 11.6 (2016), S. 881–888.
- [Kec21] KECHER, Christoph; HOFFMANN-ELBERN, Ralf and WILL, Torsten: UML 2: Das umfassende Handbuch. Bonn: Rheinwerk Computing, 2021.
- [Ken15] KENNGOTT, Hannes; WAGNER, Martin; NICKEL, Felix; WEKERLE, Anna; PREUKSCHAS, Alvin; APITZ, Mathias; SCHULTE, Thilo; REMPEL, Rudolf; MIETKOWSKI, Patrick; WAGNER, Felix u. a.: „Computer-Assisted Abdominal Surgery: New Technologies“. In: *Langenbeck’s Archives of Surgery* 400.3 (2015), S. 273–281.

- [Kir18] KIRCHNER, Kathrin and MARKOVIĆ, Petar: „Unveiling Hidden Patterns in Flexible Medical Treatment Processes – a Process Mining Case Study“. In: *International conference on decision support system technology*. Springer. Berlin, 2018, S. 169–180.
- [Kna15] KNAUER, Ulrich and KNAUER, Kolja: Diskrete und algebraische Strukturen. Berlin: Springer, 2015.
- [Kna93] KNAUF, Rainer: Logische Programmierung und wissensbasierte Systeme: Eine Einführung. Berlin: Shaker, 1993.
- [Ko09] KO, Ryan KL; LEE, Stephen SG and LEE, Eng Wah: „Business Process Management (BPM) Standards: a Survey“. In: *Business Process Management Journal* (2009).
- [Kol09] KOLLER, Daphne and FRIEDMAN, Nir: Probabilistic Graphical Models: Principles and Techniques. Cambridge: MIT Press, 2009.
- [Kol33] KOLMOGOROV, Andrei: Grundbegriffe der Wahrscheinlichkeitsrechnung. Berlin: Springer, 1933.
- [Kor10] KORB, Kevin and NICHOLSON, Ann: Bayesian Artificial Intelligence. Boston: CRC Press, 2010.
- [Kra13] KRANZFELDER, Michael; STAUB, Christoph; FIOKA, Adam; SCHNEIDER, Armin; GILLEN, Sonja; WILHELM, Dirk; FRIESS, Helmut; KNOLL, Alois and FEUSSNER, Hubertus: „Toward Increased Autonomy in the Surgical OR: Needs, Requests, and Expectations“. In: *Surgical Endoscopy* 27.5 (2013), S. 1681–1688.
- [Kre99] KREPLIN, Klaus-Dieter: The Workflow Management Coalition Specification - Konkordanz englischer und deutscher Begriffe des Workflow Management. Winchester: WFMC Publishing, 1999.
- [Kun89] KUNSCH, Hans: „The Jackknife and the Bootstrap for General Stationary Observations“. In: *The Annals of Statistics* 17.3 (1989), S. 1217–1241.
- [Kur05] KURZWEIL, Ray: The Singularity is Near: When Humans Transcend Biology. New Jersey: Penguin, 2005.

- [Kur92] KURBEL, Karl: Entwicklung und Einsatz von Expertensystemen: Eine anwendungsorientierte Einführung in wissensbasierte Systeme. Heidelberg: Springer, 1992.
- [Lal14] LALYS, Florent and JANNIN, Pierre: „Surgical Process Modelling: A Review“. In: *International Journal of Computer Assisted Radiology and Surgery* 9.3 (2014), S. 495–511.
- [Lau20] LAUE, Ralf; KOSCHMIDER, Agnes and FAHLAND, Dirk: „8 Petrinetze: Grundlagen der formalen Prozessanalyse“. In: *Prozessmanagement und Process-Mining: Grundlagen*. Berlin: De Gruyter, 2020, S. 125–152.
- [Lau98] LAUX, Helmut: „Entscheidung bei Unsicherheit im engeren Sinne“. In: *Entscheidungstheorie*. Springer, 1998, S. 103–118.
- [Lee13] LEEMANS, Sander JJ; FAHLAND, Dirk and AALST, Wil van der: „Discovering Block-structured Process Models from Event Logs - a Constructive Approach“. In: *International Conference on Applications and Theory of Petri Nets and Concurrency*. Berlin: Springer, 2013, S. 311–329.
- [Lee14] LEEMANS, Sander JJ; FAHLAND, Dirk and AALST, Wil van der: „Discovering Block-structured Process Models from Incomplete Event Logs“. In: *International Conference on Applications and Theory of Petri Nets and Concurrency*. Berlin: Springer, 2014, S. 91–110.
- [Leh19a] LEHNE, Moritz; LUIJTEN, Sandra; IMBUSCH, Paulina and THUN, Sylvia: „The Use of FHIR in Digital Health – A Review of the Scientific Literature“. In: *Studies in Health Technologies and Informatics: Vol. 9(267)* (2019), S. 52–58.
- [Leh19b] LEHNE, Moritz; SASS, Julian; ESSENWANGER, Andrea; SCHEPERS, Josef and THUN, Sylvia: „Why Digital Medicine Depends on Interoperability“. In: *NPJ Digital Medicine* 2.1 (2019), S. 1–5.
- [Leó19] LEÓN, Fernando Puente: Messtechnik: Grundlagen, Methoden und Anwendungen. Berlin: Springer, 2019.

- [Leo54] LEONARD, John Savage: The Foundations of Statistics. 1st Ed. New York: Wiley, 1954.
- [Li16] LI, Xinyu; ZHANG, Yanyi; LI, Mengzhu; CHEN, Shuhong; AUSTIN, Farneth R; MARSIC, Ivan and BURD, Randall S: „Online Process Phase Detection Using Multimodal Deep Learning“. In: *Proceedings of the Ubiquitous Computing, Electronics & Mobile Communication Conference*. Los Alamitos: IEEE Conference Publishing Services, 2016, S. 1–7.
- [Li17] LI, Xinyu; ZHANG, Yanyi; ZHANG, Jianyu; CHEN, Shuhong; MARSIC, Ivan; FARNETH, Richard A and BURD, Randall S: „Concurrent Activity Recognition with Multimodal CNN-LSTM Structure“. In: *arXiv preprint arXiv:1702.01638* (2017).
- [Lia02] LIAW, Andy; WIENER, Matthew u. a.: „Classification and Regression by RandomForest“. In: *R news* 2.3 (2002), S. 18–22.
- [Lin13] LINTE, Cristian; DAVENPORT, Katherine; CLEARY, Kevin; PETERS, Craig; VOSBURGH, Kirby; NAVAB, Nassir; JANNIN, Pierre; PETERS, Terry; HOLMES III, David R; ROBB, Richard u. a.: „On Mixed Reality Environments for Minimally Invasive Therapy Guidance: Systems Architecture, Successes and Challenges in their Implementation from Laboratory to Clinic“. In: *Computerized Medical Imaging and Graphics* 37.2 (2013), S. 83–97.
- [Lóp20] LÓPEZ-MELLADO, Ernesto and MORÁN-SOLTERO, Xitlali: „Counting Finite Transition Sequences of Block Structured Workflow Nets“. In: *IFAC-PapersOnLine* 53.4 (2020), S. 193–198.
- [Lür17] LÜRING, Christian: Zementfreie Hüftendoprothese: Lateraler Zugang nach Bauer. Berlin: Springer, 2017, S. 1–24.
- [Man16] MANDEL, Joshua; KREDA, David; MANDL, Kenneth; KOHANE, Isaac and RAMONI, Rachel: „SMART on FHIR: A Standards-based, Interoperable Apps Platform for Electronic Health Records“. In: *Journal of the American Medical Informatics Association: Vol. 23* 23 (2016), S. 899–908.

- [McC04] McCORDUCK, Pamela and CFE, Cli: *Machines who think: A personal Inquiry Into the History and Prospects of Artificial Intelligence*. New York: CRC Press, 2004.
- [McS89] McSHERRY, Charles K.: „Cholecystectomy: The Gold Standard“. In: *The American Journal of Surgery* 158.3 (1989), S. 174–178.
- [McW20] McWILLIAMS, Andrew: „Medical Robotics and Computer-Assisted Surgery: The Global Market“. In: *BCC Research HLC036H* (2020).
- [Mic21] MICROSOFT AZURE: Azure API FHIR. Online Ressource, Zugriff: 01.08.2021. 2021. URL: <https://azure.microsoft.com/en-us/services/azure-api-for-fhir/>.
- [Mol81] MOLLOY, Michael Karl: *On the Integration of Delay and Throughput Measures in Distributed Processing Models*. San Franzisko: University of California Publishing, 1981.
- [Mon20] MONETT, Dagmar; LEWIS, Colin WP; THÓRISSON, Kristinn R; BACH, Joscha; BALDASSARRE, Gianluca; GRANATO, Giovanni; BERKELEY, Istvan SN; CHOLLET, François; CROSBY, Matthew; SHEVLIN, Henry u. a.: „Special Issue “On Defining Artificial Intelligence”–Commentaries and Author’s Response“. In: *Journal of Artificial General Intelligence* 11.2 (2020), S. 1–100.
- [Mor90] MORGAN, Millett Granger; HENRION, Max and SMALL, Mitchell: *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge: Cambridge University Press, 1990.
- [Müh12] MÜHLEN, Michael: „Workflowmanagement“. In: *Prozessmanagement*. Berlin: Springer, 2012, S. 367–400.
- [Mur02] MURPHY, Kevin Patrick: „Dynamic Bayesian Networks: Representation, Inference and Learning“. Diss. University of California, Berkeley, 2002.
- [Nea04] NEAPOLITAN, Richard E: *Learning Bayesian Networks*. Bd. 38. Upper Saddle River: Prentice Hall, 2004.

- [Neu09] NEUMUTH, Thomas; JANNIN, Pierre; STRAUSS, Gero; MEIXENSBERGER, Juergen and BURGERT, Oliver: „Validation of Knowledge Acquisition for Surgical Process Models“. In: *Journal of the American Medical Informatics Association* 16.1 (2009), S. 72–80.
- [Neu12] NEUMUTH, Thomas; LIEBMANN, Peter; WIEDEMANN, Patrick and MEIXENSBERGER, Jan: „Surgical Workflow Management Schemata for Cataract Procedures“. In: *Methods of Information in Medicine* 51.05 (2012), S. 371–382.
- [New07] NEWELL, Allen and SIMON, Herbert A: „Computer Science as Empirical Inquiry: Symbols and Search“. In: *ACM Turing award Lectures*. 2007, S. 1975.
- [New76] NEWEL, Allen and SIMON, Herbert A: „Completer Science as Empirical Inquiry: Symbols and Search“. In: *Communications* (1976).
- [Nog17] NOGLER, Martin and THALER, Mathias: „Operative Zugangswege zur Hüfte beim älteren Menschen“. In: *Der Orthopäde* 46.1 (2017), S. 18–24.
- [OEC19] OECD: OECD - Health at a Glance 2019. Paris: OECD Publishing, November 2019, S. 1–220.
- [Oem21] OEMIG, Frank: „HL7 FHIR - Fast Healthcare Interoperability Resources: Eine Einführung“. In: *GMS Medizinische Informatik, Biometrie und Epidemiologie 2021: Vol. 17, No. 2* (2021).
- [OMG17] OMG: „OMG Unified Modeling Language (OMG UML) Specification Version 2.5.1“. In: *UML Specification*. Online Ressource, Zugriff: 30.03.2022. 2017. URL: <https://www.omg.org/spec/UML/2.5.1/PDF>.
- [Ort77] ORTONY, Andrew and RUMELHART, David: „The Representation of Knowledge in Memory“. In: *Schooling and the Acquisition of Knowledge* (1977), S. 99–135.

- [Pad08] PADOY, Nicolas; BLUM, Tobias; FEUSSNER, Hubertus; BERGER, Marie-Odile and NAVAB, Nassir: „On-line Recognition of Surgical Activity for Monitoring in the Operating Room“. In: *Proceedings of the Innovative Applications of Artificial Intelligence Conference*. Heidelberg: Springer, 2008, S. 1718–1724.
- [Pad10] PADOY, Nicolas: „Workflow and Activity Modeling for Monitoring Surgical Procedures“. Diss. Université Henri Poincaré-Nancy I, 2010.
- [Pat09] PATTON, Andrew; POLITIS, Dimitris N. and WHITE, Halbert: „Automatic Block-Length Selection for the Dependent Bootstrap“. In: *Econometric Reviews* 28.4 (2009), S. 372–375.
- [Pea88] PEARL, Judea: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. New York: Morgan Kaufmann, 1988.
- [Pér94] PÉRISSAT, Jacques: „Laparoskopische Cholezystektomie“. In: *Operationslehre der Endoskopischen Chirurgie 1*. Hrsg. von BUESS, Gerhard; CUSCHIERI, Alfred and PÉRISSAT, Jacques. Heidelberg: Springer, 1994, S. 225–250.
- [Pet62] PETRI, Carl Adam: „Kommunikation mit Automaten“. Diss. Universität Bonn, 1962.
- [Pet67] PETRI, Carl Adam: „Gundsätzliches zur Beschreibung diskreter Prozesse“. In: *Kolloquium über Automatentheorie* 3 (1967), S. 121–140.
- [Phi15a] PHILIPP, Patrick; ECK, Ralf and PEINSIPP-BYMA, Elisabeth: „Decision Support in Context of a Standard-based Usability Evaluation“. In: *Proceedings of the Global Online Conference on Information and Computer Technology, November 4-6, Louisville, Kentucky, USA*. Los Alamitos: IEEE Conference Publishing Service, 2015, S. 36–41.
- [Phi15b] PHILIPP, Patrick; FISCHER, Yvonne; HEMPEL, Dirk and BEYERER, Jürgen: „Framework for an Interactive Assistance in Diagnostic Processes Based on the Translation of UML Activities into Petri

Nets“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, December 7-9, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2015, S. 732–737.

- [Phi15c] PHILIPP, Patrick; FISCHER, Yvonne; HEMPEL, Dirk and BEYERER, Jürgen: „Modeling of Clinical Practice Guidelines for Interactive Assistance in Diagnostic Processes“. In: *Proceedings of the World Congress in Computer Science, Computer Engineering, and Applied Computing, International Conference on Health Informatics and Medical Systems, July 27-30, Las Vegas, Nevada, USA*. Athens: CS-REA Press, 2015, S. 3–9.
- [Phi16a] PHILIPP, Patrick: „Framework for Modeling Medical Guidelines Based on the Translation of UML Activities into YAWL“. In: *Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, July 19-26, Triberg-Nussbach, Germany*. Karlsruhe: KIT Scientific Publishing, 2016, S. 1–16.
- [Phi16b] PHILIPP, Patrick; FISCHER, Yvonne; HEMPEL, Dirk and BEYERER, Jürgen: „Framework for an Interactive Assistance in Diagnostic Processes Based on Probabilistic Modeling of Clinical Practice Guidelines“. In: *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*. Amsterdam: Morgan Kaufmann, 2016, S. 371–390.
- [Phi16c] PHILIPP, Patrick; SCHREITER, Luzie; GIEHL, Johannes; FISCHER, Yvonne; RACZKOWSKY, Joerg; SCHWARZ, Markus; WOERN, Heinz and BEYERER, Jürgen: „Situation Detection for an Interactive Assistance in Surgical Interventions Based on Dynamic Bayesian Networks“. In: *Proceedings of the 6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery, September 14-16, Pisa, Italy*. Pisa: Scuola Superiore Sant’Anna, 2016, S. 45–46.

- [Phi17a] PHILIPP, Patrick: „Combining YAWL and DBNs for Surgical Phase Detection“. In: *Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, July 24-29, Triberg-Nussbach, Germany*. Karlsruhe: KIT Scientific Publishing, 2017, S. 1–16.
- [Phi17b] PHILIPP, Patrick; BLEIER, Johannes; FISCHER, Yvonne and BEYERER, Jürgen: „Towards a surgical phase detection using Markov Logic Networks“. In: *Proceedings of the 17th Annual Meeting of the International Society for Computer Assisted Orthopaedic Surgery, June 14-17, Aachen, Germany*. Manchester: EasyChair EPiC Series in Health Sciences, 2017, S. 288–294.
- [Phi17c] PHILIPP, Patrick and FISCHER, Yvonne: „Vorrichtung und Verfahren zur Bestimmung eines Zustandes eines Arbeitsablaufs“. In: *Eingereichte deutsche Patentanmeldung Fraunhofer-Gesellschaft und Karlsruher Institut für Technologie*. München, 2017, S. 1–52.
- [Phi17d] PHILIPP, Patrick; FISCHER, Yvonne and BEYERER, Jürgen: „Expert-based Probabilistic Modeling of Workflows in Context of Surgical Interventions“. In: *Proceedings of the IEEE Conference on Cognitive and Computational Aspects of Situation Management, March 27-31, Savannah, Georgia, USA*. Los Alamitos: IEEE Conference Publishing Services, 2017, S. 77–83.
- [Phi18a] PHILIPP, Patrick: „Phase Detection in Medical Context: Overview and Outlook“. In: *Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, July 30 - August 05, Triberg-Nussbach, Germany*. Karlsruhe: KIT Scientific Publishing, 2018, S. 63–81.
- [Phi18b] PHILIPP, Patrick; BOMMERSHEIM, Marie; ROBERT, Sebastian; HEMPEL, Dirk and BEYERER, Jürgen: „Modeling of Clinical Practice Guidelines for an Interactive Decision Support Using Ontologies“. In: *Proceedings of the World Congress in Computer Science, Computer Engineering, and Applied Computing, International Conference on Health Informatics and Medical Systems, July 30 -*

August 02, Las Vegas, Nevada, USA. Athens: CSREA Press, 2018, S. 83–89.

- [Phi18c] PHILIPP, Patrick and FISCHER, Yvonne: „Vorrichtung und Verfahren zur Bestimmung eines Zustandes eines Arbeitsablaufs“. In: *Eingereichte PCT Patentanmeldung Fraunhofer-Gesellschaft und Karlsruher Institut für Technologie*. München, 2018, S. 1–45.
- [Phi18d] PHILIPP, Patrick and ROBERT, Sebastian: „Sensible Roboter sollen erkennen, was der Mensch will“. In: *PC-Welt Wissenschaft und Forschung: Robotics*. Online Ressource, Zugriff: 08.04.2020. München: IDG Tech Media GmbH, 2018. URL: <https://www.pcwelt.de/a/sensible-roboter-sollen-erkennen-was-der-mensch-will>.
- [Phi18e] PHILIPP, Patrick; ROBERT, Sebastian and BEYERER, Jürgen: „Interactive Decision Support: A Framework to Improve Diagnostic Processes of Cancerous Diseases Using Bayesian Networks“. In: *Proceedings of the IEEE Conference on Cognitive and Computational Aspects of Situation Management, June 11-14, Boston, Massachusetts, USA*. Los Alamitos: IEEE Conference Publishing Services, 2018, S. 15–21.
- [Phi19a] PHILIPP, Patrick; BECKER, Silvia; ROBERT, Sebastian and BEYERER, Jürgen: „Modeling of Medical Treatment Processes for an Interactive Assistance Based on the Translation of UML Activities into PROforma“. In: *Proceedings of 6th Intl. Conference on Computational Science and Computational Intelligence, December 05-07, Las Vegas, Nevada, USA*. Athens: CSREA Press, 2019, S. 50–55.
- [Phi19b] PHILIPP, Patrick; BOMMERSHEIM, Marie; ROBERT, Sebastian and BEYERER, Jürgen: „Probabilistic Estimation of Human Interaction Needs in Context of a Robotic Assistance in Geriatrics“. In: *Current Directions in Biometrical Engineering*. Berlin: De Gruyter, 2019, S. 433–436.
- [Phi19c] PHILIPP, Patrick; MORALES GEORGI, Rafael Xavier; ROBERT, Sebastian and BEYERER, Jürgen: „Analysis of Control Flow Graphs Using Graph Convolutional Neural Networks“. In: *Proceedings of 6th Intl. IEEE Conference on Soft Computing and Machine Learning*,

- November 19-20, Johannesburg, South Africa. Red Hook: Curran Associates Inc., 2019, S. 73–77.
- [Phi20] PHILIPP, Patrick; JACOB, Ruben; ROBERT, Sebastian and BEYERER, Jürgen: „Predictive Analysis of Business Processes Using Neural Networks with Attention Mechanism“. In: *Proceedings of the Intl. Conference on Artificial Intelligence in Information and Communication, February 19-21, Fukuoka, Japan*. Los Alamitos: IEEE Conference Publishing Services, 2020, S. 225–230.
- [Phi21] PHILIPP, Patrick; HEMPEL, Louisa; HEMPEL, Dirk and BEYERER, Jürgen: „Modeling Clinical Practice Guidelines for Interactive Decision Support Exemplified by Primary Myelofibrosis and Immune Thrombocytopenia“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 15-17, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2021, S. 523–528.
- [Phi22] PHILIPP, Patrick and BOMMERSHEIM, Marie: „Event Abstraction in Medical Context Using Self Organizing Maps (derzeit im Druck)“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 12-14, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2022.
- [Plo19] PLONER, Nico; NEURATH, Markus; SCHOENTHALER, Martin; ANDREAS, Zielke and PROKOSCH, Hans: „Concept to Gain Trust for a German Personal Health Record System using Public Cloud and FHIR“. In: *Journal of Biomedical Informatics: Vol.95(101312)* (2019), S. 1–11.
- [Pol04] POLITIS, Dimitris and WHITE, Halbert: „Automatic Block-Length Selection for the Dependent Bootstrap“. In: *Econometric Reviews* 23.1 (2004), S. 53–70.

- [Pol91] POLITIS, Dimitris and ROMANO, Joseph: A Circular Block-resampling Procedure for Stationary Data. West Lafayette: Purdue University, Department of Statistics, 1991.
- [Pre09] PRESS, James: Subjective and Objective Bayesian Statistics: Principles, Models, and Applications. Bd. 590. New Jersey: Wiley, 2009.
- [Pro93] PROSSER, Alexander: „Das ISO-OSI-Modell“. In: *Standards in Rechnernetzen*. Heidelberg: Springer, 1993, S. 2–7.
- [Psc13] PSCHYREMBEL, Willibald; ARNOLD, Ulrike; BILIC, Michaela Viktorija; GROßMANN, Alexander and WEDI, Bettina: Pschyrembel Klinisches Wörterbuch (2014). 265., überarb. Aufl. München: De Gruyter, 2013.
- [Pup13a] PUPPE, Frank: Einführung in Expertensysteme. Heidelberg: Springer, 2013.
- [Pup13b] PUPPE, Frank: Problemlösungsmethoden in Expertensystemen. Heidelberg: Springer, 2013.
- [Rab86] RABINER, Lawrence and JUANG, B: „An Introduction to Hidden Markov Models“. In: *IEE ASSP Magazine* 3.1 (1986), S. 4–16.
- [REF15] REFA: Industrial Engineering - Standardmethoden zur Produktivitätssteigerung und Prozessoptimierung. Darmstadt: REFA Verlag, 2015.
- [Rei13] REISIG, Wolfgang: Understanding Petri Nets. Heidelberg: Springer, 2013.
- [Rok10] ROKACH, Lior: Pattern Classification Using Ensemble Methods. Bd. 75. Singapur: World Scientific Publishing, 2010.
- [Rom19] ROMANOWSKI, Josh: „Self Organizing Maps als Werkzeug bei der Erkennung von Arbeitsabläufen“. Bachelorarbeit. Karlsruher Institut für Technologie, 2019.
- [Rup12] RUPP, Chris and QUEINS, Stefan: UML 2 glasklar. München: Carl Hanser Verlag GmbH Co KG, 2012.

- [Rus09] RUSSELL, Stuart and NORVIG, Peter: *Artificial Intelligence: A Modern Approach*. 3rd. New York: Prentice Hall Press, 2009.
- [Sch16] SCHREITER, Luzie; PHILIPP, Patrick; GIEHL, Johannes; FISCHER, Yvonne; SCHWARZ, Markus; RACZKOWSKY, Joerg; BEYERER, Jürgen and WOERN, Heinz: „Situation Detection for an Interactive Assistance in Surgical Interventions Based on Random Forests“. In: *Proceedings of the International Journal of Computer Assisted Radiology and Surgery, International Congress and Exhibition Computer Assisted Radiology and Surgery, June 21-25, Heidelberg, Germany*. Heidelberg: Springer, 2016, S. 115–116.
- [Sea80] SEARLE, John: „Minds, Brains, and Programs“. In: *Behavioral and Brain Sciences* 3.3 (1980), S. 417–424.
- [Sel14] SELL, John and O’CONNOR, Patrick: „The Xbox One System on a Chip and Kinect Sensor“. In: *IEEE Micro* 34.2 (2014), S. 44–53.
- [She19] SHEPPARD, Kevin: Arch Introduction. Online Ressource, Zugriff: 07.07.2021. 2019. URL: <https://arch.readthedocs.io/en/latest/index.html>.
- [Smo87] SMOLENSKY, Paul: „Connectionist AI, symbolic AI, and the brain“. In: *Artificial Intelligence Review* 1.2 (1987), S. 95–109.
- [Smo90] SMOLENSKY, Paul: „Tensor Product Variable Binding and the Representation of Symbolic Structures in Connectionist Systems“. In: *Artificial intelligence* 46.1-2 (1990), S. 159–216.
- [Sta14] STAUDER, Ralf; OKUR, Asli; PETER, Loïc; SCHNEIDER, Armin; KRANZFELDER, Michael; FEUSSNER, Hubertus and NAVAB, Nassir: „Random Forests for Phase Detection in Surgical Workflow Analysis“. In: *Proceedings of the International Conference on Information Processing in Computer-Assisted Interventions*. Springer. Heidelberg, 2014, S. 148–157.
- [Sta19a] STATISTISCHES BUNDESAMT: *Anzahl der Implantationen künstlicher Hüftgelenke in deutschen Krankenhäusern in den Jahren 2005 bis 2018*. Bonn: GBE Verlag, Oktober 2019, S. 5.

- [Sta19b] STATISTISCHES BUNDESAMT: Häufigste Operationen an vollstationär behandelten Krankenhauspatienten in Deutschland in den Jahren 2016 bis 2018. Bonn: GBE Verlag, September 2019, S. 1.
- [Sta73] STACHOWIAK, Herbert: Allgemeine Modelltheorie. Heidelberg: Springer, 1973.
- [Ste11] STEINBERG, Earl; GREENFIELD, Sheldon and MANCHER, Michelle: Clinical Practice Guidelines We Can Trust. Washington: National Academies Press, 2011.
- [Ste65] STEINBUCH, Karl: „Automat und Mensch“. In: *Automat und Mensch*. Springer, 1965, S. 390–411.
- [Stö05] STÖRRLE, Harald: UML 2 für Studenten. Bd. 320. New York: Pearson Studium, 2005.
- [Thu21] THUN, Sylvia: „Anwendungen von FHIR in der digitalen Medizin“. In: *GMS Medizinische Informatik, Biometrie und Epidemiologie 2021: Vol. 17, No. 2* (2021).
- [Tin76] TINHOFER, Gottfried: Methoden der angewandten Graphentheorie. Berlin: Springer, 1976.
- [Tit19] TITTMANN, Peter: Graphentheorie: Eine Anwendungsorientierte Einführung. 3. Aufl. München: Hanser Verlag, 2019.
- [Tra76] TRAVERSO, William: „Carl Langenbuch and the First Cholecystectomy“. In: *The American Journal of Surgery* 132.1 (1976), S. 81–82.
- [Tre91] TREDE, M; SAEGER, H-D; SCHAUPP, W and PETERMANN, C: „Laparoskopische Cholezystektomie“. In: *DMW-Deutsche Medizinische Wochenschrift* 116.06 (1991), S. 219–222.
- [Tur09] TURING, Alan: „Computing Machinery and Intelligence“. In: *Parsing the Turing Test*. Berlin: Springer, 2009, S. 23–65.
- [Tur50] TURING, Alan: „Computing Machinery and Intelligence“. In: *Mind* 59.236 (1950), S. 433–460.

- [Twi17] TWINANDA, Andru P; SHEHATA, Sherif; MUTTER, Didier; MARESCAUX, Jacques; MATHELIN, Michel de and PADOY, Nicolas: „Endonet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos“. In: *IEEE Journal Transactions on Medical Imaging* 36.1 (2017), S. 86–97.
- [Van04] VAN DER AALST, Wil and VAN HEE, Kees: *Workflow Management: Models, Methods, and Systems*. Boston: MIT Press, 2004.
- [Van86] VAN GULIK, TM: „Langenbuch’s Cholecystectomy, Once a Remarkably Controversial Operation.“ In: *The Netherlands Journal of Surgery* 38.5 (1986), S. 138.
- [van96] VAN DER AALST, Wil: „Three Good Reasons for Using a Petri-Net-Based Workflow Management System“. In: *Proceedings of the International Working Conference on Information and Process Integration in Enterprises*. Singapur: World Scientific Publishing, 1996, S. 179–201.
- [van98] VAN DER AALST, Wil: „The Application of Petri Nets to Workflow Management“. Englisch. In: *Journal of Circuits, Systems, and Computers* 8.01 (1998), S. 21–66.
- [Ven66] VENN, John: *The Logic of Chance: An Essay on the Foundations and Province of the Theory of Probability, with Special Reference to Its Application to Moral and Social Science*. London: Macmillan, 1866.
- [Vos96] VOSSEN, Gottfried and BECKER, Jörg: *Geschäftsprozessmodellierung und Workflow-Management: Modelle, Methoden, Werkzeuge*. Informatik-Lehrbuch-Reihe. New York: International Thomson Publishing, 1996.
- [Wat86] WATERMAN, Donald: *A Guide to Expert Systems*. Reading: Addison-Wesley, 1986.
- [Wes12] WESKE, Mathias: *Business Process Management*. Berlin: Springer, 2012.
- [Wie48] WIENER, Norbert: *Cybernetics or Control and Communication in the Animal and the Machine*. New York: Technology Press, 1948.

- [Wil16] WILKINSON, Mark D; DUMONTIER, Michel; AALBERSBERG, IJsbrand Jan; APPLETON, Gabrielle; AXTON, Myles; BAAK, Arie; BLOMBERG, Niklas; BOITEN, Jan-Willem; SILVA SANTOS, Luiz Bonino da and BOURNE, Philip: „The FAIR Guiding Principles for Scientific Data Management and Stewardship“. In: *Scientific data* 3.1 (2016), S. 1–9.
- [Win67] WINKLER, Robert: „The Assessment of Prior Distributions in Bayesian Analysis“. In: *Journal of the American Statistical Association* 62.319 (1967), S. 776–800.
- [Wöl05] WÖLNERHANSEN, Bettina; ACKERMANN, Camilla; GUENIN, Marc and KERN, Beatrice: „Zwölf Jahre laparoskopische Cholezystektomie“. In: *Der Chirurg* 76.3 (2005), S. 263–269.
- [Wor99] WORKFLOW MANAGEMENT COALITION: The Workflow Management Coalition Specification - Terminology and Glossary (Document Number WFMC-TC-1011). Winchester: WFMC Publishing, 1999.
- [Yu18] YU, Kun-Hsing; BEAM, Andrew and KOHANE, Isaac: „Artificial Intelligence in Healthcare“. In: *Nature Biomedical Engineering* 2.10 (2018), S. 719–731.
- [Zel20] ZELST, Sebastiaan van and LEEMANS, Sander: „Translating Workflow Nets to Process Trees: An Algorithmic Approach“. In: *Algorithms* 13.11 (2020), S. 279.
- [Zho12] ZHOU, Zhi-Hua: *Ensemble Methods: Foundations and Algorithms*. Boca Raton: CRC Press, 2012.

Eigene Publikationen

- [Phi15a] PHILIPP, Patrick; ECK, Ralf and PEINSIPP-BYMA, Elisabeth: „Decision Support in Context of a Standard-based Usability Evaluation“. In: *Proceedings of the Global Online Conference on Information and Computer Technology, November 4-6, Louisville, Kentucky, USA*. Los Alamitos: IEEE Conference Publishing Service, 2015, S. 36–41.
- [Phi15b] PHILIPP, Patrick; FISCHER, Yvonne; HEMPEL, Dirk and BEYERER, Jürgen: „Framework for an Interactive Assistance in Diagnostic Processes Based on the Translation of UML Activities into Petri Nets“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, December 7-9, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2015, S. 732–737.
- [Phi15c] PHILIPP, Patrick; FISCHER, Yvonne; HEMPEL, Dirk and BEYERER, Jürgen: „Modeling of Clinical Practice Guidelines for Interactive Assistance in Diagnostic Processes“. In: *Proceedings of the World Congress in Computer Science, Computer Engineering, and Applied Computing, International Conference on Health Informatics and Medical Systems, July 27-30, Las Vegas, Nevada, USA*. Athens: CS-REA Press, 2015, S. 3–9.
- [Phi16a] PHILIPP, Patrick: „Framework for Modeling Medical Guidelines Based on the Translation of UML Activities into YAWL“. In: *Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, July 19-26,*

Triberg-Nussbach, Germany. Karlsruhe: KIT Scientific Publishing, 2016, S. 1–16.

- [Phi16b] PHILIPP, Patrick; FISCHER, Yvonne; HEMPEL, Dirk and BEYERER, Jürgen: „Framework for an Interactive Assistance in Diagnostic Processes Based on Probabilistic Modeling of Clinical Practice Guidelines“. In: *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*. Amsterdam: Morgan Kaufmann, 2016, S. 371–390.
- [Phi16c] PHILIPP, Patrick; SCHREITER, Luzie; GIEHL, Johannes; FISCHER, Yvonne; RACZKOWSKY, Joerg; SCHWARZ, Markus; WOERN, Heinz and BEYERER, Jürgen: „Situation Detection for an Interactive Assistance in Surgical Interventions Based on Dynamic Bayesian Networks“. In: *Proceedings of the 6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery, September 14-16, Pisa, Italy*. Pisa: Scuola Superiore Sant’Anna, 2016, S. 45–46.
- [Sch16] SCHREITER, Luzie; PHILIPP, Patrick; GIEHL, Johannes; FISCHER, Yvonne; SCHWARZ, Markus; RACZKOWSKY, Joerg; BEYERER, Jürgen and WOERN, Heinz: „Situation Detection for an Interactive Assistance in Surgical Interventions Based on Random Forests“. In: *Proceedings of the International Journal of Computer Assisted Radiology and Surgery, International Congress and Exhibition Computer Assisted Radiology and Surgery, June 21-25, Heidelberg, Germany*. Heidelberg: Springer, 2016, S. 115–116.
- [Phi17a] PHILIPP, Patrick: „Combining YAWL and DBNs for Surgical Phase Detection“. In: *Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, July 24-29, Triberg-Nussbach, Germany*. Karlsruhe: KIT Scientific Publishing, 2017, S. 1–16.
- [Phi17b] PHILIPP, Patrick; BLEIER, Johannes; FISCHER, Yvonne and BEYERER, Jürgen: „Towards a surgical phase detection using Markov Logic Networks“. In: *Proceedings of the 17th Annual Meeting of the International Society for Computer Assisted Orthopaedic Surgery*,

June 14-17, Aachen, Germany. Manchester: Easychair EPiC Series in Health Sciences, 2017, S. 288–294.

- [Phi17c] PHILIPP, Patrick and FISCHER, Yvonne: „Vorrichtung und Verfahren zur Bestimmung eines Zustandes eines Arbeitsablaufs“. In: *Eingereichte deutsche Patentanmeldung Fraunhofer-Gesellschaft und Karlsruher Institut für Technologie.* München, 2017, S. 1–52.
- [Phi17d] PHILIPP, Patrick; FISCHER, Yvonne and BEYERER, Jürgen: „Expert-based Probabilistic Modeling of Workflows in Context of Surgical Interventions“. In: *Proceedings of the IEEE Conference on Cognitive and Computational Aspects of Situation Management, March 27-31, Savannah, Georgia, USA.* Los Alamitos: IEEE Conference Publishing Services, 2017, S. 77–83.
- [Phi18a] PHILIPP, Patrick: „Phase Detection in Medical Context: Overview and Outlook“. In: *Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory, July 30 - August 05, Triberg-Nussbach, Germany.* Karlsruhe: KIT Scientific Publishing, 2018, S. 63–81.
- [Phi18b] PHILIPP, Patrick; BOMMERSHEIM, Marie; ROBERT, Sebastian; HEMPEL, Dirk and BEYERER, Jürgen: „Modeling of Clinical Practice Guidelines for an Interactive Decision Support Using Ontologies“. In: *Proceedings of the World Congress in Computer Science, Computer Engineering, and Applied Computing, International Conference on Health Informatics and Medical Systems, July 30 - August 02, Las Vegas, Nevada, USA.* Athens: CSREA Press, 2018, S. 83–89.
- [Phi18c] PHILIPP, Patrick and FISCHER, Yvonne: „Vorrichtung und Verfahren zur Bestimmung eines Zustandes eines Arbeitsablaufs“. In: *Eingereichte PCT Patentanmeldung Fraunhofer-Gesellschaft und Karlsruher Institut für Technologie.* München, 2018, S. 1–45.
- [Phi18d] PHILIPP, Patrick and ROBERT, Sebastian: „Sensible Roboter sollen erkennen, was der Mensch will“. In: *PC-Welt Wissenschaft und*

Forschung: Robotics. Online Ressource, Zugriff: 08.04.2020. München: IDG Tech Media GmbH, 2018. URL: <https://www.pcwelt.de/a/sensible-roboter-sollen-erkennen-was-der-mensch-will>.

- [Phi18e] PHILIPP, Patrick; ROBERT, Sebastian and BEYERER, Jürgen: „Interactive Decision Support: A Framework to Improve Diagnostic Processes of Cancerous Diseases Using Bayesian Networks“. In: *Proceedings of the IEEE Conference on Cognitive and Computational Aspects of Situation Management, June 11-14, Boston, Massachusetts, USA*. Los Alamitos: IEEE Conference Publishing Services, 2018, S. 15–21.
- [Phi19a] PHILIPP, Patrick; BECKER, Silvia; ROBERT, Sebastian and BEYERER, Jürgen: „Modeling of Medical Treatment Processes for an Interactive Assistance Based on the Translation of UML Activities into PROforma“. In: *Proceedings of 6th Intl. Conference on Computational Science and Computational Intelligence, December 05-07, Las Vegas, Nevada, USA*. Athens: CSREA Press, 2019, S. 50–55.
- [Phi19b] PHILIPP, Patrick; BOMMERSHEIM, Marie; ROBERT, Sebastian and BEYERER, Jürgen: „Probabilistic Estimation of Human Interaction Needs in Context of a Robotic Assistance in Geriatrics“. In: *Current Directions in Biometrical Engineering*. Berlin: De Gruyter, 2019, S. 433–436.
- [Phi19c] PHILIPP, Patrick; MORALES GEORGI, Rafael Xavier; ROBERT, Sebastian and BEYERER, Jürgen: „Analysis of Control Flow Graphs Using Graph Convolutional Neural Networks“. In: *Proceedings of 6th Intl. IEEE Conference on Soft Computing and Machine Learning, November 19-20, Johannesburg, South Africa*. Red Hook: Curran Associates Inc., 2019, S. 73–77.
- [Phi20] PHILIPP, Patrick; JACOB, Ruben; ROBERT, Sebastian and BEYERER, Jürgen: „Predictive Analysis of Business Processes Using Neural Networks with Attention Mechanism“. In: *Proceedings of the Intl. Conference on Artificial Intelligence in Information and Communication, February 19-21, Fukuoka, Japan*. Los Alamitos: IEEE Conference Publishing Services, 2020, S. 225–230.

- [Bom21a] BOMMERSHEIM, Marie and PHILIPP, Patrick: „Supporting the Incorporation of Individual Patient Preferences for Decision Support in Breast Cancer Therapy“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 15-17, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2021, S. 523–525.
- [Bom21b] BOMMERSHEIM, Marie; PHILIPP, Patrick; BEYERER, Jürgen and HEMPEL, Dirk: „Workflow Analysis for Supporting Breast Cancer Therapy Using Process Mining“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, July 26-29, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2021, S. 523–528.
- [Hem21a] HEMPEL, Louisa; VELOSO DE OLIVEIRA, Julia; GAUMANN, Andreas; PHILIPP, Patrick; GAARG, Arun; ROBERT, Sebastian and HEMPEL, Dirk: „Landscape of Biomarkers and Actionable Gene Alterations in Adenocarcinoma of GEJ and Stomach“. In: *Zeitschrift für Gastroenterologie* 59.08 (2021).
- [Hem21b] HEMPEL, Louisa; VELOSO DE OLIVEIRA, Julia; GAUMANN, Andreas; PHILIPP, Patrick; MEDERLE, Stefanie; GAARG, Arun; PIEHLER, Armin; KLEESPIES, Axel; ROBERT, Sebastian and HEMPEL, Dirk: „Landscape of Biomarkers and Actionable Gene Alterations in Adenocarcinoma of GEJ and Stomach—A Real World Data Analysis“. In: *Cancers* 13.17 (2021).
- [Phi21] PHILIPP, Patrick; HEMPEL, Louisa; HEMPEL, Dirk and BEYERER, Jürgen: „Modeling Clinical Practice Guidelines for Interactive Decision Support Exemplified by Primary Myelofibrosis and Immune Thrombocytopenia“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 15-17, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2021, S. 523–528.

- [App22a] APPENZELLER, Arno; LEITNER, Moritz; PHILIPP, Patrick; KREMPEL, Erik and BEYERER, Jürgen: „Privacy and Utility of Private Synthetic Data for Medical Data Analyses“. In: *Journal of Applied Sciences* 12.23 (2022), S. 1–19.
- [App22b] APPENZELLER, Arno; TERZER, Nick Moritz; PHILIPP, Patrick and BEYERER, Jürgen: „Applying Differential Privacy to Medical Questionnaires (derzeit im Review)“. In: *Proceedings of the IEEE First Workshop on Privacy Preserving Computation in Pervasive Computing, März 13-17, Atlanta, Georgia, USA*. Los Alamitos: IEEE Conference Publishing Services, 2022.
- [Bom22] BOMMERSHEIM, Marie and PHILIPP, Patrick: „Quality Criteria Analysis in Process Models Discovered for Breast Cancer Therapy“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, July 25-28, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2022, S. 514–519.
- [Bra22] BRAHIM, Mouna; WESTERKAMP, Kai; HEMPEL, Louisa; LEHMANN, Reiner; HEMPEL, Dirk and PHILIPP, Patrick: „Automated Assessment of Breast Positioning Quality in Screening Mammography“. In: *Cancers* 14.4704 (2022), S. 1–29.
- [Hem22a] HEMPEL, Louisa; GAUMANN, Andreas; VELOSO, Julia; PHILIPP, Patrick; HEMPEL, Dirk; ROBERT, Sebastian and EBNER, Florian: „Zielgerichtete Therapien für das Endometriumkarzinom im Real-World-Setting“. In: *Geburtshilfe und Frauenheilkunde* 82.10 (2022), S. 139.
- [Hem22b] HEMPEL, Louisa; LAPA, Constantin; GAUMANN, Andreas; SCHEIBER, Josef; VELOSO, Julia; PHILIPP, Patrick; ROBERT, Sebastian and HEMPEL, Dirk: „The Point Mutation p.C382R in the Transmembrane Domain of FGFR2 is Oncogenic and Leads to a Complete

Functional Remission after Treatment with Pemigatinib in Cholangiocarcinoma“. In: *Poster in Proceedings of the Targeted Anticancer Therapies (TAT) Congress, März 7-9, Lugano, Schweiz*. Lugano: ESMO Publications, 2022, S. 60.

- [Hem22c] HEMPEL, Louisa; PIEHLER, Armin; GANDORFER, Beate; VELOSO, Julia; PHILIPP, Patrick; ROBERT, Sebastian and HEMPEL, Dirk: „A new Promising Oncogenic Target (p.C382R) for Treatment with Pemigatinib in Patients with Cholangiocarcinoma“. In: *Therapeutic Advances in Medical Oncology* 22.04 (2022), S. 1–10.
- [Hem22d] HEMPEL, Louisa; PIEHLER, Armin; GANDORFER, Beate; VELOSO, Julia; PHILIPP, Patrick; ROBERT, Sebastian and HEMPEL, Dirk: „Clinical Impact of SARS-CoV-2 Delta Variant Infection in Tumor Patients and the Impact of Vaccination on Different Cancer Treatment Regimens“. In: *Journal of Clinical Oncology* 40.16 (2022), S. 70–83.
- [Hem22e] HEMPEL, Louisa; VELOSO, Julia; SCHEIBER, Josef; PHILIPP, Patrick; ROBERT, Sebastian and HEMPEL, Dirk: „A point mutation replacing Cysteine with Arginine at position 382 (C382R) in the transmembrane domain of FGFR2 leads to response to FGF2-inhibitor Pemigatinib in chemo-refractory intrahepatic cholangiocarcinoma“. In: *Proceedings of the Targeted Anticancer Therapies (TAT) Congress, März 7-9, Lugano, Schweiz*. Lugano: ESMO Publications, 2022, S. 89–93.
- [Phi22a] PHILIPP, Patrick and BOMMERSHEIM, Marie: „Event Abstraction in Medical Context Using Self Organizing Maps (derzeit im Druck)“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 12-14, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2022.
- [Phi22b] PHILIPP, Patrick; VELOSO, Julia; APPENZELLER, Arno and BEYERER, Jürgen: „Evaluation of an Automated Mapping from ICD-10

to SNOMED CT (derzeit im Druck)“. In: *Proceedings of the International Conference on Computational Science and Computational Intelligence, International Symposium on Health Informatics and Medical Systems, Dezember 12-14, Las Vegas, Nevada, USA*. Los Alamitos: IEEE Conference Publishing Services, 2022.

- [Rad22] RADIC, Marija; BUSCH-CASLER, Julia; VOSEN, Agnes; APPENZELER, Arno; MUCHA, Henrik; PHILIPP, Patrick; FRANK, Kevin; DAUTH, Stephanie; KOEHM, Michaela and ORAK: „Data Sovereignty for Patient-oriented Clinical Research in Germany: Requirements from a Clinical, Technical, Legal and Economic Perspective (derzeit im Review)“. In: *SAGE Digital Health (2022)*.

Betreute studentische Arbeiten

- [Ach19] ACHOUR, Mélék: „Modelling of Medical Workflows with Bayesian Networks on the Application Example of Hip Replacement Surgery“. Bachelorarbeit. Karlsruher Institut für Technologie, 2019.
- [Bec19] BECKER, Silvia: „Programmierung einer interaktiven Benutzeroberfläche für die Auswertung von Immunantworten“. Praktikumsarbeit. Karlsruher Institut für Technologie, 2019.
- [Ble16] BLEIER, Johannes: „Abgleich medizinischer Arbeitsabläufe mit Markov Logik Netzen“. Seminararbeit. Karlsruher Institut für Technologie, 2016.
- [Bom16] BOMMERSHEIM, Marie: „Abgleich von Arbeitsabläufen im Interventionsraum“. Proseminararbeit. Karlsruher Institut für Technologie, 2016.
- [Bom17] BOMMERSHEIM, Marie: „Modellierung medizinischer Leitlinien für eine interaktive Unterstützung bei Therapieprozessen“. Bachelorarbeit. Karlsruher Institut für Technologie, 2017.
- [Bom20] BOMMERSHEIM, Marie: „Analyse von Arbeitsabläufen für die Unterstützung von Diagnose und Therapie bei Brustkrebs“. Masterarbeit. Karlsruher Institut für Technologie, 2020.
- [Bra21] BRAHIM, Mouna: „Automatisierte Bewertung der Brustpositionierung in der Mammographie“. Masterarbeit. Karlsruher Institut für Technologie, 2021.
- [Dre17] DREINER, Johannes Michael: „Phasenerkennung und Anomalieerkennung in medizinischen Arbeitsabläufen mittels CNN-LSTM Netzwerken“. Diplomarbeit. Karlsruher Institut für Technologie, 2017.

- [Fra18] FRANZ, Maximilian: „Scalable Inductive Process Mining“. Proseminararbeit. Karlsruher Institut für Technologie, 2018.
- [Hag19] HAGMANN, Raphael: „Automatisierte Generierung von Prozessmodellen mittels Self Organizing Maps“. Seminararbeit. Karlsruher Institut für Technologie, 2019.
- [Jac19] JACOB, Ruben: „Prädiktive Analyse von Geschäftsprozessen durch Neuronale Netze mit Attention-Mechanismen“. Bachelorarbeit. Karlsruher Institut für Technologie, 2019.
- [Kol17] KOLNER, Oleh: „Künstliche Intelligenz zum Abgleich von Workflowphasen“. Proseminararbeit. Karlsruher Institut für Technologie, 2017.
- [Kol18] KOLNER, Oleh: „Recognition of Workflow Phases Using Differentiable Neural Computers on the Application Example of an Interactive Assistance“. Bachelorarbeit. Karlsruher Institut für Technologie, 2018.
- [Kot16a] KOTYRBA, Fabian Kevin: „Situationserkennung im Operationsaal“. Proseminararbeit. Karlsruher Institut für Technologie, 2016.
- [Kot16b] KOTYRBA, Fabian Kevin: „Workflowmodellierung mit Petri-Netzen am Beispiel einer interaktiven Unterstützung im Interventionsraum“. Bachelorarbeit. Karlsruher Institut für Technologie, 2016.
- [Leo21] LEO, Klaudia: „Probabilistic Modeling of Examination Results in the Context of Breast Cancer Diagnostics“. Bachelorarbeit. Karlsruher Institut für Technologie, 2021.
- [Mai18] MAIER, Melina: „Affective Computing mit Bayes-Netzen“. Praktikumsarbeit. Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB), 2018.
- [Mei21] MEISTER, Daniel: „Genomischen Analysen und Wirkmechanismen in Kombination mit Cluster-Verfahren und Process-Mining“. Praktikumsarbeit. Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB), 2021.

- [Mor20] MORALES GEORGI, Rafael Xavier: „Graphkonvolutionale Neuronale Netzen für das unüberwachte Lernen von Kontrollflussgraphen in Arbeitsabläufen“. Bachelorarbeit. Karlsruher Institut für Technologie, 2020.
- [Mül17] MÜLLER, Florian: „Modellierung räumlich-zeitlicher Daten mit künstlichen neuronalen Netzen am Beispiel medizinischer Arbeitsabläufe“. Masterarbeit. Karlsruher Institut für Technologie, 2017.
- [Mum18] MUMPER, Robert: „Entdecken von Arbeitsabläufen“. Proseminararbeit. Karlsruher Institut für Technologie, 2018.
- [Ohl18] OHLSSON, Tobias Lars: „Vorausschauende Geschäftsprozessüberwachung durch neuronale Netze mit Gated Recurrent Units“. Bachelorarbeit. Karlsruher Institut für Technologie, 2018.
- [Oli20] OLIVEIRA, Julia de: „Datenbereinigung und -analyse zum kolorektalen Karzinom“. Praktikumsarbeit. Fraunhofer Institut für Optonik, Systemtechnik und Bildauswertung (IOSB), 2020.
- [Pöh17] PÖHLER, Fabian: „Workflow-Assistent im Interventionsraum“. Proseminararbeit. Karlsruher Institut für Technologie, 2017.
- [Rob21] ROBAK, Linda: „Clusteranalyse von onkologischen Realdaten“. Masterarbeit. Karlsruher Institut für Technologie, 2021.
- [Rom19] ROMANOWSKI, Josh: „Self Organizing Maps als Werkzeug bei der Erkennung von Arbeitsabläufen“. Bachelorarbeit. Karlsruher Institut für Technologie, 2019.
- [Sch18] SCHÜTT, Anan: „Automated Recognition of Workflow Phases via Surgery Videos using Autoencoder and LSTM“. Bachelorarbeit. Karlsruher Institut für Technologie, 2018.
- [Sek21] SEKBAN, Ildeniz: „Predicting Chemotherapy Regimens from Electronic Health Record Time Series Data of Colorectal Cancer Patients Using Machine Learning Techniques“. Masterarbeit. Karlsruher Institut für Technologie, 2021.
- [Sip18] SIPPPEL, David: „Abgleich und Prädiktion von Arbeitsabläufen“. Seminararbeit. Karlsruher Institut für Technologie, 2018.

- [Taf19] TAFUR, Santiago: „Workflow Discovery“. Proseminararbeit. Karlsruher Institut für Technologie, 2019.
- [Vet18] VETTER, Dennis: „Methoden zum datengestützten Erkennen von Arbeitsabläufen - Ein Überblick“. Seminararbeit. Karlsruher Institut für Technologie, 2018.

Karlsruher Schriftenreihe zur Anthropomatik (ISSN 1863-6489)

- Band 1** Jürgen Geisler
Leistung des Menschen am Bildschirmarbeitsplatz.
ISBN 3-86644-070-7
- Band 2** Elisabeth Peinsipp-Byma
Leistungserhöhung durch Assistenz in interaktiven Systemen zur Szenenanalyse. 2007
ISBN 978-3-86644-149-1
- Band 3** Jürgen Geisler, Jürgen Beyerer (Hrsg.)
Mensch-Maschine-Systeme.
ISBN 978-3-86644-457-7
- Band 4** Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2009 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-469-0
- Band 5** Thomas Usländer
Service-oriented design of environmental information systems.
ISBN 978-3-86644-499-7
- Band 6** Giulio Milighetti
Multisensorielle diskret-kontinuierliche Überwachung und Regelung humanoider Roboter.
ISBN 978-3-86644-568-0
- Band 7** Jürgen Beyerer, Marco Huber (Hrsg.)
Proceedings of the 2010 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-609-0
- Band 8** Eduardo Monari
Dynamische Sensorselektion zur auftragsorientierten Objektverfolgung in Kameranetzwerken.
ISBN 978-3-86644-729-5

- Band 9** Thomas Bader
Multimodale Interaktion in Multi-Display-Umgebungen.
ISBN 3-86644-760-8
- Band 10** Christian Frese
Planung kooperativer Fahrmanöver für kognitive Automobile.
ISBN 978-3-86644-798-1
- Band 11** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2011 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-855-1
- Band 12** Miriam Schleipen
Adaptivität und Interoperabilität von Manufacturing Execution Systemen (MES).
ISBN 978-3-86644-955-8
- Band 13** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2012 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-86644-988-6
- Band 14** Hauke-Hendrik Vagts
Privatheit und Datenschutz in der intelligenten Überwachung: Ein datenschutzgewährendes System, entworfen nach dem „Privacy by Design“ Prinzip.
ISBN 978-3-7315-0041-4
- Band 15** Christian Kühnert
Data-driven Methods for Fault Localization in Process Technology. 2013
ISBN 978-3-7315-0098-8
- Band 16** Alexander Bauer
Probabilistische Szenenmodelle für die Luftbildauswertung.
ISBN 978-3-7315-0167-1
- Band 17** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2013 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0212-8

- Band 18** Michael Teutsch
Moving Object Detection and Segmentation for Remote Aerial Video Surveillance.
ISBN 978-3-7315-0320-0
- Band 19** Marco Huber
Nonlinear Gaussian Filtering: Theory, Algorithms, and Applications.
ISBN 978-3-7315-0338-5
- Band 20** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2014 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0401-6
- Band 21** Todor Dimitrov
Permanente Optimierung dynamischer Probleme der Fertigungssteuerung unter Einbeziehung von Benutzerinteraktionen.
ISBN 978-3-7315-0426-9
- Band 22** Benjamin Kühn
Interessengetriebene audiovisuelle Szenenexploration.
ISBN 978-3-7315-0457-3
- Band 23** Yvonne Fischer
Wissensbasierte probabilistische Modellierung für die Situationsanalyse am Beispiel der maritimen Überwachung.
ISBN 978-3-7315-0460-3
- Band 24** Jürgen Beyerer, Alexey Pak (Hrsg.)
Proceedings of the 2015 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0519-8
- Band 25** Pascal Birnstill
Privacy-Respecting Smart Video Surveillance Based on Usage Control Enforcement.
ISBN 978-3-7315-0538-9
- Band 26** Philipp Woock
Umgebungskartenschätzung aus Sidescan-Sonar-daten für ein autonomes Unterwasserfahrzeug.
ISBN 978-3-7315-0541-9

- Band 27** Janko Petereit
Adaptive State × Time Lattices: A Contribution to Mobile Robot Motion Planning in Unstructured Dynamic Environments.
ISBN 978-3-7315-0580-8
- Band 28** Erik Ludwig Krempel
Steigerung der Akzeptanz von intelligenter Videoüberwachung in öffentlichen Räumen.
ISBN 978-3-7315-0598-3
- Band 29** Jürgen Moßgraber
Ein Rahmenwerk für die Architektur von Frühwarnsystemen. 2017
ISBN 978-3-7315-0638-6
- Band 30** Andrey Belkin
World Modeling for Intelligent Autonomous Systems.
ISBN 978-3-7315-0641-6
- Band 31** Chettapong Janya-Anurak
Framework for Analysis and Identification of Nonlinear Distributed Parameter Systems using Bayesian Uncertainty Quantification based on Generalized Polynomial Chaos.
ISBN 978-3-7315-0642-3
- Band 32** David Münch
Begriffliche Situationsanalyse aus Videodaten bei unvollständiger und fehlerhafter Information.
ISBN 978-3-7315-0644-7
- Band 33** Jürgen Beyerer, Alexey Pak (Eds.)
Proceedings of the 2016 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0678-2
- Band 34** Jürgen Beyerer, Alexey Pak and Miro Taphanel (Eds.)
Proceedings of the 2017 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0779-6
- Band 35** Michael Grinberg
Feature-Based Probabilistic Data Association for Video-Based Multi-Object Tracking.
ISBN 978-3-7315-0781-9

- Band 36** Christian Herrmann
Video-to-Video Face Recognition for Low-Quality Surveillance Data.
ISBN 978-3-7315-0799-4
- Band 37** Chengchao Qu
Facial Texture Super-Resolution by Fitting 3D Face Models.
ISBN 978-3-7315-0828-1
- Band 38** Miriam Ruf
Geometrie und Topologie von Trajektorienoptimierung für vollautomatisches Fahren.
ISBN 978-3-7315-0832-8
- Band 39** Angelika Zube
Bewegungsregelung mobiler Manipulatoren für die Mensch-Roboter-Interaktion mittels kartesischer modellprädiktiver Regelung.
ISBN 978-3-7315-0855-7
- Band 40** Jürgen Beyerer and Miro Taphanel (Eds.)
Proceedings of the 2018 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-0936-3
- Band 41** Marco Thomas Gewohn
Ein methodischer Beitrag zur hybriden Regelung der Produktionsqualität in der Fahrzeugmontage.
ISBN 978-3-7315-0893-9
- Band 42** Tianyi Guan
Predictive energy-efficient motion trajectory optimization of electric vehicles.
ISBN 978-3-7315-0978-3
- Band 43** Jürgen Metzler
Robuste Detektion, Verfolgung und Wiedererkennung von Personen in Videodaten mit niedriger Auflösung.
ISBN 978-3-7315-0968-4
- Band 44** Sebastian Bullinger
Image-Based 3D Reconstruction of Dynamic Objects Using Instance-Aware Multibody Structure from Motion.
ISBN 978-3-7315-1012-3

- Band 45** Jürgen Beyerer, Tim Zander (Eds.)
Proceedings of the 2019 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-1028-4
- Band 46** Stefan Becker
Dynamic Switching State Systems for Visual Tracking.
ISBN 978-3-7315-1038-3
- Band 47** Jennifer Sander
Ansätze zur lokalen Bayes'schen Fusion von Informationsbeiträgen heterogener Quellen.
ISBN 978-3-7315-1062-8
- Band 48** Philipp Christoph Sebastian Bier
Umsetzung des datenschutzrechtlichen Auskunftsanspruchs auf Grundlage von Usage-Control und Data-Provenance-Technologien.
ISBN 978-3-7315-1082-6
- Band 49** Thomas Emter
Integrierte Multi-Sensor-Fusion für die simultane Lokalisierung und Kartenerstellung für mobile Robotersysteme.
ISBN 978-3-7315-1074-1
- Band 50** Patrick Dunau
Tracking von Menschen und menschlichen Zuständen.
ISBN 978-3-7315-1086-4
- Band 51** Jürgen Beyerer, Tim Zander (Eds.)
Proceedings of the 2020 Joint Workshop of Fraunhofer IOSB and Institute for Anthropomatics, Vision and Fusion Laboratory.
ISBN 978-3-7315-1091-8
- Band 52** Lars Wilko Sommer
Deep Learning based Vehicle Detection in Aerial Imagery.
ISBN 978-3-7315-1113-7
- Band 53** Jan Hendrik Hammer
Interaktionstechniken für mobile Augmented-Reality-Anwendungen basierend auf Blick- und Handbewegungen.
ISBN 978-3-7315-1169-4

- Band 54** Jürgen Beyerer, Tim Zander (Eds.)
**Proceedings of the 2021 Joint Workshop of
Fraunhofer IOSB and Institute for Anthropomatics,
Vision and Fusion Laboratory.**
ISBN 978-3-7315-1171-7
- Band 55** Ronny Hug
Probabilistic Parametric Curves for Sequence Modeling.
ISBN 978-3-7315-1198-4
- Band 56** Florian Patzer
**Automatisierte, minimalinvasive Sicherheitsanalyse und
Vorfalreaktion für industrielle Systeme.**
ISBN 978-3-7315-1207-3
- Band 57** Achim Christian Kuwertz
**Adaptive Umweltmodellierung für kognitive Systeme in
offener Welt durch dynamische Konzepte und quantitative
Modellbewertung.**
ISBN 978-3-7315-1219-6
- Band 58** Julius Pfrommer
Distributed Planning for Self-Organizing Production Systems.
ISBN 978-3-7315-1253-0
- Band 59** Ankush Meshram
Self-learning Anomaly Detection in Industrial Production.
ISBN 978-3-7315-1257-8
- Band 60** Patrick Philipp
**Über die Formalisierung und Analyse medizinischer Prozesse
im Kontext von Expertenwissen und künstlicher Intelligenz.**
ISBN 978-3-7315-1289-9

Lehrstuhl für Interaktive Echtzeitsysteme
Karlsruher Institut für Technologie

Fraunhofer-Institut für Optronik, Systemtechnik
und Bildauswertung IOSB Karlsruhe

Die Digitalisierung hat bereits viele Bereiche der Wirtschaft und des gesellschaftlichen Lebens verändert. Auch unterliegen die Aspekte des Gesundheitswesens und der klinischen Praxis einem digitalen Wandel. Im Hinblick auf diese Entwicklungen beleuchtet die vorliegende Arbeit die Akquisition, Repräsentation und Nutzung von Prozesswissen im Kontext hybrider KI-Methoden. Ein Beitrag ist die strukturerhaltende Hin- und Rücktransformation von Prozessbäumen zu Prozessplänen. Eine probabilistische Erweiterung von Prozessplänen sowie eine Transformation im Bereich einer Problemlösungskomponente runden die Beiträge ab.

ISSN 1863-6489
ISBN 978-3-7315-1289-9

