



 **Universität Trier**

Modellierung, Visualisierung und Ausführung
wissensintensiver Prozesse unter Verwendung
Semantischer Technologien

Dissertation
am Fachbereich IV der Universität Trier
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

Eric Rietzke
(Dipl. Informatiker FH)

Berichterstatter: Prof. Dr. Ralph Bergmann
Prof. Dr. Norbert Kuhn
Dekan: Prof. Dr. Ludwig von Auer
Eingereicht am: 12.09.2021
Disputation am: 05.11.2021

Eric Rietzke:
*Modellierung, Visualisierung und Ausführung wissensintensiver Prozesse
unter Verwendung Semantischer Technologien* © 2021.

Danksagung

Im Verlauf der letzten sechs Jahre wurde diese Dissertation am Lehrstuhl für Wirtschaftsinformatik II der Universität Trier in Kooperation mit der Hochschule Trier - Umwelt-Campus Birkenfeld durchgeführt.

An erster Stelle möchte ich mich bei meinem Doktorvater Prof. Dr. Ralph Bergmann für seine ständige Unterstützung und seine Inspiration im gesamten Promotionsverlauf bedanken. Seine umfassende fachliche Expertise zu Geschäftsprozessen sowie semantischen Technologien hat meine thematische Ausrichtung und mein Wissen stark beeinflusst. Für meine Forschung sowie für die Fertigstellung dieser Arbeit waren seine konstruktiven Ratschläge unverzichtbar.

Des Weiteren gilt mein Dank Prof. Dr. Norbert Kuhn, der mich bei der Initiierung und Durchführung der Promotion immer unterstützt hat. Sein Feedback und die Ideen aus unseren zahlreichen Gesprächen waren wertvolle Impulse auf dem Weg meiner Forschung. Trotz seiner Tätigkeit als Präsident der Hochschule Trier fand er immer die Zeit zum Austausch.

Zudem möchte ich mich bei allen Kollegen und auch Studenten bedanken, die mich auf unterschiedlichste Weise im Verlauf meiner Promotion unterstützt haben. Im Besonderen gilt mein Dank Dr. Markus Schwinn, Lisa Grumbach, Carsten Maletzki und Valentin Kächele für die vielen Momente des gegenseitigen Austauschs, der Zusammenarbeit bei Publikationen sowie der Implementierung einzelner Module und frühen Demonstratoren.

Darüber hinaus wäre die Ausarbeitung des Anwendungsszenarios im Leitstellenumfeld nicht ohne die freundliche Unterstützung der Mitarbeiter des DRKs und der Integrierten Leitstelle in Ludwigshafen möglich gewesen. Hier gilt mein Dank Hans-Peter Adolph für die Unterstützung und das Engagement über die letzten Jahre sowie Dr. Fred Blaschke für die medizinische sowie leitstellenbezogene Expertise beim experimentellen Aufbau einer erweiterten Notrufabfrage. Ein Dankeschön auch an alle Disponenten der Leitstelle Ludwigshafen für die Mitwirkung im Rahmen der Evaluation.

Meiner Familie bin ich von Herzen dankbar, dass sie mich auf diesem Weg so liebevoll begleitet und motiviert hat, um im Gegenzug dabei viele Stunden und Wochenenden auf mich verzichten zu müssen. Dies gilt im Besonderen für meine Frau Heidi Rietzke, "who has silently corrected my grammar".

Es war eine anstrengende aber auch aufregende Phase, an die ich sicher noch lange zurück denken werde.

Eric Rietzke
Trier, September 2021

Zusammenfassung

In vielen Branchen und vor allem in großen Unternehmen gehört eine Unterstützung von Geschäftsprozessen durch Workflow-Management-Systeme zum gelebten Alltag. Im Zentrum steht dabei die Steuerung kontrollflussorientierter Abläufe, während Prozesse mit einem Schwerpunkt auf Daten, Informationen und Wissen meist außen vor bleiben. Solche wissensintensive Prozesse (engl.: knowledge intensive processes) (KiPs) sind Untersuchungsgegenstand in vielen aktuellen Studien, welche ein derzeit aktives Forschungsgebiet formen.

Im Vordergrund solcher KiPs steht dabei das durch die mitwirkenden Personen eingebrachte Wissen, welches in einem wesentlichen Maß die Prozessausführung beeinflusst, hierdurch jedoch die Bearbeitung komplexer und meist hoch volatiler Prozesse ermöglicht. Hierbei handelt es sich zumeist um entscheidungsintensive Prozesse, Prozesse zur Wissensakquisition oder Prozesse, die zu einer Vielzahl unterschiedlicher Prozessabläufe führen können.

Im Rahmen dieser Arbeit wird ein Ansatz entwickelt und vorgestellt, der sich der Modellierung, Visualisierung und Ausführung wissensintensiver Prozesse unter Verwendung Semantischer Technologien widmet. Hierzu werden als die zentralen Anforderungen zur Ausführung von KiPs Flexibilität, Adaptivität und Zielorientierung definiert. Daran anknüpfend werden drei zentrale Grundprinzipien der Prozessmodellierung identifiziert, welche in der ersten Forschungsfrage aufgegriffen werden: „Können die drei Grundprinzipien in einem einheitlichen datenzentrierten, deklarativen, semantischen Ansatz (welcher mit ODD-BP bezeichnet wird) kombiniert werden und können damit die zentralen Anforderungen von KiPs erfüllt werden?“

Die Grundlage für ODD-BP bildet ein Metamodell, welches als Sprachkonstrukt fungiert und die Definition der angestrebten Prozessmodelle erlaubt. Darauf aufbauend wird mit Hilfe von Inferenzierungsregeln ein Verfahren entwickelt, welches das Schlussfolgern von Prozesszuständen ermöglicht und somit eine klassische Workflow-Engine überflüssig macht. Zudem

wird eine Methodik eingeführt, die für jede in einem Prozess mitwirkende Person eine maßgeschneiderte, adaptive Prozessvisualisierung ermöglicht, um neben dem Freiheitsgrad der Flexibilität auch eine fundierte Prozessunterstützung bei der Ausführung von KiPs leisten zu können. All dies erfolgt innerhalb einer einheitlichen Wissensbasis, die zum einen die Grundlage für eine vollständige semantische Prozessmodellierung bildet und zum anderen die Möglichkeit zur Integration von Expertenwissen eröffnet. Dieses Expertenwissen kann einen expliziten Beitrag bei der Ausführung wissensintensiver Prozesse leisten und somit die Kollaboration von Mensch und Maschine durch Technologien der symbolischen KI ermöglichen. Die zweite Forschungsfrage greift diesen Aspekt auf: „Kann in dem ODD-BP Ansatz ontologisches Wissen so integriert werden, dass dieses in einer Prozessausführung einen Beitrag leistet?“

Das Metamodell sowie die entwickelten Methoden und Verfahren werden in einem prototypischen, generischen System realisiert, welches grundsätzlich für alle Anwendungsgebiete mit KiPs geeignet ist. Zur Validierung des ODD-BP Ansatzes erfolgt eine Ausrichtung auf den Anwendungsfall einer Notrufabfrage aus dem Leitstellenumfeld. Im Zuge der Evaluation wird gezeigt, wie dieser wissensintensive Ablauf von einer flexiblen, adaptiven und zielorientierten Prozessausführung profitiert. Darüber hinaus wird medizinisches Expertenwissen in den Prozessablauf integriert und es wird nachgewiesen, wie dieses zu verbesserten Prozessergebnissen beiträgt.

Wissensintensive Prozesse stellen Unternehmen und Organisationen in allen Branchen und Anwendungsfällen derzeit vor große Herausforderungen und die Wissenschaft und Forschung widmet sich der Suche nach praxistauglichen Lösungen. Diese Arbeit präsentiert mit ODD-BP einen vielversprechenden Ansatz, indem die Möglichkeiten Semantischer Technologien dazu genutzt werden, eine eng verzahnte Zusammenarbeit zwischen Mensch und Maschine bei der Ausführung von KiPs zu ermöglichen. Die zur Evaluation fokussierte Notrufabfrage innerhalb von Leitstellen stellt zudem einen höchst relevanten Anwendungsfall dar, da in einem akuten Notfall in kürzester Zeit Entscheidungen getroffen werden müssen, um weitreichenden Schaden abwenden und Leben retten zu können. Durch die Berücksichtigung umfassender Datenmengen und das Ausnutzen verfügbaren Expertenwissens kann so eine schnelle Lagebewertung mit Hilfe der maschinellen Unterstützung erreicht und der Mensch beim Treffen von richtigen Entscheidungen unterstützt werden.

Summary

In many industries and especially in large companies, supporting business processes with workflow management systems are part of everyday life. The focal point is mainly on control-flow-oriented processes, while processes focusing on data, information, and knowledge are usually left out. Such knowledge-intensive processes (KiPs) are the subject of investigation in many current studies, which form an active field of research.

In the foreground of such KiPs, there is the knowledge brought in by the persons involved, which influences the execution of a process to a significant degree, but on the other hand enables the handling of complex and often highly volatile processes. These are mostly decision-intensive processes, processes for knowledge acquisition, or processes that can lead to many different process flows.

In the context of this thesis, an approach is developed and presented, which is dedicated to the modeling, visualization, and execution of knowledge-intensive processes by using semantic technologies. For this purpose, the major requirements to execute KiPs are defined as flexibility, adaptivity, and goal orientation. Based on this, three fundamental principles of process modeling are identified, which are addressed with the first research question: „Can the three fundamental principles be combined into a unified data-centric, declarative, semantic approach (which will be labeled ODD-BP) and thus meet the central requirements of KiPs?"

The foundation for ODD-BP is a metamodel that acts as a language construct and allows the definition of the envisioned process models. Based on this, a procedure is developed using inferencing rules, which allow to deduce process states and thus makes a classical workflow engine superfluous. In addition, a methodology is introduced to provide a customized adaptive process visualization for each person involved in a process, thus providing not only the degree of freedom of flexibility, but also process support for the execution of KiPs. All this is done within a unified knowledge base, which forms the basis for a complete semantic process modeling on the

one hand and, on the other hand, opens the possibility to integrate expert knowledge. This expert knowledge directly contributes to the execution of knowledge-intensive processes, thus enabling human-machine collaboration through symbolic AI technologies. The second research question addresses this aspect: „Can ontological knowledge be integrated into the ODD-BP approach so that it contributes to a process execution?"

The metamodel as well as the developed methods and procedures will be implemented in a prototypical generic system, which is basically suitable for all application areas with KiPs. For the validation of the ODD-BP approach, the use case of an emergency call is focussed within the environment of an emergency control center. The evaluation course shows how this knowledge-intensive process benefits from a flexible, adaptive, and goal-oriented process execution. Furthermore, medical expert knowledge is integrated into the process flow and it is demonstrated how this contributes to improved process outcomes.

Knowledge-intensive processes challenge companies and organizations in all industries while academia and research are dedicated to find practical solutions. This work presents ODD-BP, a promising approach by leveraging the power of semantic technologies to enable tightly integrated human-machine collaboration in the execution of KiPs. The emergency call procedure within an emergency control center represents a highly relevant use case, since in an acute emergency, decisions have to be made in the shortest possible time to avert far-reaching damage and save lives. By taking into account extensive amounts of data and exploiting available expert knowledge, a rapid assessment of a situation can be achieved with the help of machine support, while humans can be assisted in making the right decisions.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	2
1.2	Ziele der Dissertation	4
1.3	Forschungsmethodik	7
1.4	Forschungsprojekte	9
1.4.1	SEMAFLEX	9
1.4.2	SEMANAS	10
1.5	Generelles zur Dissertation	10
1.5.1	Die Zielgruppe	10
1.5.2	Geschlechterneutrale Formulierungen	11
1.5.3	Hervorhebungen	11
1.6	Aufbau der Dissertation	11
2	Grundlagen	13
2.1	Daten - Informationen - Wissen	13
2.2	Prozessorientierte Informationssysteme	15
2.3	Geschäftsprozesse	16
2.3.1	Terminologie	18
2.3.2	Grundprinzipien der Prozessmodellierung	21
2.4	Prozessmodellierungssprachen	25
2.5	Charakteristiken von Geschäftsprozessen	28
2.5.1	Flexibilität	28
2.5.2	Adaptivität	31
2.5.3	Zielorientierung	33
2.6	Wissensintensive Prozesse	34
2.6.1	Definition von KiPs	35
2.6.2	Charakteristiken von KiPs	35
2.6.3	Anforderungen von KiPs	37
2.6.4	Forschungslücke	39
2.7	Semantische Technologien	40
2.7.1	Begrifflichkeiten	40
2.7.2	Arten einer Formalisierung	41
2.7.3	RDF / RDFS	42

2.7.4	OWL	45
2.7.5	SWRL	46
2.7.6	Schlussfolgerungen	47
2.8	Semantische Prozessmodellierung	48
2.8.1	SBPM-Lifecycle	49
2.8.2	Auszug relevanter Arbeiten	52
2.8.3	Forschungslücken	53
2.9	Prozessvisualisierung	55
3	Problemstellung und Lösungsansatz	61
3.1	Kernanforderungen von KiPs	61
3.2	Lösungsansatz	62
3.3	Artefakte	64
3.3.1	A1: Metamodell	64
3.3.2	A2: Inferenzieren von Prozesszuständen	64
3.3.3	A3: Adaptive Prozessvisualisierung	65
3.3.4	A4: ODD-BP System	66
3.4	Bezug zu den Forschungsfragen	67
4	Anwendungsszenario	69
4.1	Motivation	70
4.2	Der Anwendungsfall	71
4.3	Die Anwender:innen	71
4.4	SNA - Repräsentativ für KiPs	73
4.5	Stand der Technik	76
4.6	Fiktiver Notruf	79
4.7	Defizite	80
4.8	Fazit	82
5	Konzeption des Metamodells	83
5.1	Systemarchitektur	83
5.2	Metamodell	85
5.2.1	Dataobjects und Attributes	89
5.2.2	Tasks und Ressourcen	89
5.3	Prozessbeispiel unter Verwendung des Metamodells	91
5.4	Fazit	92
6	Inferenzierung von Prozesszuständen	93
6.1	Inferenzierung der Ausführbarkeit	95
6.2	Inferenzierung der Ausführungsrelevanz	98

6.3	Inferenzierung der Zielrelevanz	100
6.4	Logic-Task und Condition-States	102
6.5	Inferenzierung von Process Goals	105
6.6	Herausforderungen durch die Open World Assumption	106
6.7	Prozessunterstützung mittels Expertenwissen	107
6.8	Demonstration der Inferenzierungsmethodik	108
6.9	Fazit	111
7	Adaptive Prozessvisualisierung	113
7.1	Verfahren der Prozessvisualisierung	116
7.2	Grundlagen einer Prozessvisualisierung	117
7.3	Generelle Methoden der Modifikation	121
7.3.1	Abstraktion	122
7.3.2	Reduktion	122
7.3.3	Aggregation	123
7.3.4	Expansion	124
7.4	Adaptionsverfahren	126
7.4.1	Kontext	126
7.4.2	POIs	127
7.4.3	Propagation	127
7.4.4	Modifikation	130
7.5	Erweiterung der Ontologie	131
7.6	Beispiel eines Adaptionsvorgangs	134
7.7	Fazit	137
8	ODD-BP System	139
8.1	Systemarchitektur	139
8.1.1	Verwendete Technologien	141
8.1.2	Basisfunktionalität	143
8.2	Modellierung des Domänenwissens für die Anwendung	148
8.2.1	Domainspezifische Konzepte	148
8.2.2	KI-Support Konzepte und Regeln	149
8.3	Prozessmodelle des Anwendungsszenarios	153
8.4	System Demonstration	159
8.4.1	Prozessstart	159
8.4.2	Prozessinstanz nach bSNA und vor eSNA	161
8.4.3	Prozessinstanz nach eSNA	163
9	Evaluation	167
9.1	Hypothesen	167

9.2	Experimenteller Aufbau	169
9.2.1	Simulierte Notfälle	170
9.2.2	Fragenkatalog	175
9.2.3	Ablauf eines Testdurchlaufs	176
9.2.4	Teilnehmer:innen	178
9.3	Ergebnisse zu Hypothese 1	180
9.4	Ergebnisse zu Hypothese 2	184
9.5	Ergebnisse zu Hypothese 3	188
10	Fazit	193
10.1	Zusammenfassung	193
10.2	Wissenschaftlicher Beitrag	195
10.3	Evaluationsergebnisse	197
10.4	Zukünftige Arbeiten	200
10.5	Perspektiven	202

Abbildungsverzeichnis

2.1	Varianten von PAIS nach Dumas et al. [33, S. 15]	16
2.2	Lebenszyklus eines Geschäftsprozesses	18
2.3	Prozesszustände: Imperativer versus deklarativer Modellierungsansatz in Anlehnung an [84]	22
2.4	Flexibilität versus Unterstützung nach [84]	23
2.5	Differenzierung der Prozessflexibilität in Anlehnung an [116] .	29
2.6	OWL-Statement in RDF/XML Syntax	45
2.7	SBPM-Lebenszyklus [123]	50
2.8	Freiheitsgrade der Gestaltung einer Prozessvisualisierung [20]	56
3.1	Übergeordneter Rahmen dieser Arbeit	67
4.1	Basis-Segment der SNA-Methodik (Integrierten Leitstelle der Stadt Ludwigshafen (ILtS-Ludwigshafen))	72
5.1	Prozessrelevantes Wissen und der Einfluss auf eine Prozessausführung	84
5.2	Auszug aus dem Metamodell der Base-Ontologie	86
5.3	Auszug Task/Ressourcen Beziehungen aus der Base-Ontologie	90
5.4	Prozessinstanz auf Basis des vorgestellten Metamodells	91
6.1	Inferenzierungskonzept	93
6.2	Erweiterung der Base-Ontologie	94
6.3	Beispiel mit <i>Execution-States</i> und <i>Relevance-States</i>	101
6.4	Logic Tasks extension of the Base-ontology	103
6.5	Beispiel mit einem <i>Condition</i> Element	104
6.6	Beispiel mit Darstellung der Inferenzierungsmechanismen . .	109
7.1	Verfahren der Prozessvisualisierung und Interaktion	116
7.2	Transfer von drei Task-Prozesselementen in eine PV	117
7.3	Einführung weiterer Prozesselement-Typen	119
7.4	Freiheitsgrade der Gestaltung einer Prozessvisualisierung . . .	121
7.5	Darstellung eines Datenelements in drei Abstraktionsstufen .	122
7.6	Darstellung zwei unterschiedlicher Aggregationsmöglichkeiten	123

7.7	Darstellung einer Expansion	124
7.8	Verfahren einer adaptierten Prozessvisualisierung	126
7.9	Erweiterung der Ontology um anwenderspezifische Parameter	131
7.10	Schichtenmodell der anwenderspezifischen Parameter	133
7.11	Adaption einer orginären PV zu einer aPV	134
8.1	Datentransfer und Verarbeitungsschritte	140
8.2	Ausschnitt einer Prozessdefinition	145
8.3	Vollständiges Prozessmodell der bSNA und eSNA	154
8.4	Handlungen	155
8.5	Daten	156
8.6	bSNA	157
8.7	eSNA	158
8.8	aPV einer Prozessinstanz nach Eingabe der ersten Frage . . .	160
8.9	aPV einer Prozessinstanz nach Durchlauf gemäß bSNA	162
8.10	aPV einer Prozessinstanz nach Durchlauf inkl. eSNA	164
9.1	Gemessene durchschnittliche Variabilität $Var(S)$ für jeden Prozessschritt S	182

Tabellenverzeichnis

4.2	Fiktiver Notruf: Sturz	79
9.1	Übersicht der Testteilnehmer:innen inkl. Testpattern und individuellem Profil	179
9.2	Anzahl der jeweiligen Benotung zu jeder Frage inkl. Mittelwert	185
9.3	Dispositionsergebnisse für Fall A: Bewusstlosigkeit	189
9.4	Dispositionsergebnisse für Fall B: Schmerzen Brustbereich . .	190
9.5	Dispositionsergebnisse für Fall C: Verkehrsunfall	191

1 Einleitung

In Gesellschaft und Wissenschaft begegnet man dem Begriff *Prozess* in unterschiedlichen Variationen und das jeweilige Verständnis wird geprägt durch die individuellen Erfahrungen aus diversen Anwendungsbereichen. Die Herstellung von Schießpulver oder die Raffination von Rohstoffen wie Erdöl erfolgen durch einen chemischen Prozess. Henry Ford implementierte 1908 erstmals die Fließbandproduktion in der Automobilindustrie und veränderte das Verständnis für Fertigungsprozesse. In den 50ern und 60ern des 20. Jahrhunderts etablierten sich Container für den Transport von Gütern und gestatteten wesentlich effizientere und schnellere Logistikprozesse, welche sich weltweit als neuer Standard durchsetzten.

In jeder Branche und jedem Anwendungsgebiet finden sich solche Prozesse und eine Unterstützung und Steuerung durch ein computergestütztes System zur Verbesserung von quantitativen sowie qualitativen Merkmalen ist ein Ziel der Wirtschaftsinformatik. [82, 12] In diesem Kontext wird der Begriff *Geschäftsprozess* geprägt, abgeleitet von dem formalen Prozessbegriff wie er in der Informatik verwendet wird [101]. Nach DIN EN ISO 9000:2015 ist ein *Geschäftsprozess* ein „Satz zusammenhängender oder sich gegenseitig beeinflussender Tätigkeiten, der Eingaben zum Erzielen eines vorgesehenen Ergebnisses verwendet.“ Im Fokus stehen demnach Tätigkeiten und deren Vernetzung; ein Leitbild, das auf die meisten Geschäftsprozesse angewendet werden kann und dabei die Organisation von Aktivitäten in den Vordergrund stellt.

Mit der flächendeckenden Verbreitung von Computern in Unternehmen entwickelte sich auch der Bedarf zur computergestützten Steuerung von Geschäftsprozessen, der zunächst durch proprietäre Lösungen einzelner Anbieter adressiert wurde [5]. Aus den Erfahrungen zur strukturierten Definition von Programmabläufen mittels der Unified Modeling Language (UML) erwuchs in den letzten 20 Jahren eine standardisierte Form zur Beschreibung von Geschäftsprozessen mit der Bezeichnung Business Process Model and Notation (BPMN) [81], welche heute den Quasi-Standard zur Modellierung von Geschäftsprozessen darstellt. Dem etablierten Verständnis von Geschäftsprozessen folgend und das imperative Prinzip von UML aufgreifend hat BPMN einen entsprechend überwiegend kontrollflussorientierten

Charakter ausgeprägt. Einem Ansatz wie BPMN folgend wird dann dieser Charakter auch von den damit umgesetzten Geschäftsprozessmodellen reflektiert, was ein spezielles Informationsmodell zur Repräsentation von Geschäftsprozessen [12] kennzeichnet.

1.1 Motivation

Der gesellschaftliche Wandel vom Industriezeitalter hin zum Informationszeitalter [26, S. 1-4] führt auch zu Veränderungen in etablierten Geschäftsprozessen und verlangt nach entsprechenden Anpassungen der Unternehmen [51][82, S. 1-6]. Insbesondere der unmittelbare Einfluss von Daten, Informationen und Wissen auf sich verändernde und sich neu entwickelnde Prozesse rückt in den Fokus wissenschaftlicher Untersuchungen [8, 42]. Gronau und Weber [43] bezeichnen diese Art von Prozessen als wissensintensive Prozesse (engl.: knowledge intensive processes) (KiPs) und definieren dazu: „Ein Prozess ist wissensintensiv, wenn sein Wert nur durch die Erfüllung der Wissensanforderungen der Prozessteilnehmer geschaffen werden kann.“

Im Zentrum wissenschaftlicher Arbeiten zu KiPs liegt dabei vorwiegend der Einfluss von Daten, Informationen und Wissen auf den Prozessverlauf und den daraus resultierenden Anforderungen an ein Geschäftsprozessmodell. Ausgangspunkt dieser Bemühungen ist die diesen Arbeiten zugrundeliegende Überzeugung, dass der imperative und kontrollflussorientierte Charakter des BPMN-Standards nicht geeignet ist, um wissensintensive Prozesse mit ihren besonderen Charakteristiken und Anforderungen zielgerichtet ausführen zu können. Van der Aalst et. al. adressieren in diesem Kontext die Notwendigkeit, die Möglichkeiten aber auch die Herausforderungen von Adaptionen innerhalb von Geschäftsprozessen [114, 113]. Mit ADEPT2 [89], Guard-Stage-Milestone [28] und Case Management [23] werden Ansätze entwickelt und vorgestellt, welche sich gezielt der Veränderlichkeit von Prozessen widmen. Die Charakteristiken und Anforderungen wissensintensiver Prozesse werden von Di Ciccio [32] mit Bezug auf die Vorarbeiten zu KiPs zusammenfassend dargestellt. Darin werden auch die unterschiedlichsten etablierten Ansätze hinsichtlich ihrer Eignung für KiPs untersucht und es wird festgestellt, dass noch weitreichende Defizite bestehen. Kapitel 2.6 widmet sich diesem Thema im Detail und zeigt hierzu eine zentrale Forschungslücke auf, welche im Rahmen dieser Arbeit adressiert wird.

In einem Umfeld, in dem Daten, Information, Wissen eine zentrale Bedeutung zugeschrieben wird, liegt es nahe, sich mit darauf ausgerichteten Methoden und Technologien auseinanderzusetzen. Hierbei kommt Informationssystemen [39] die Aufgabe der Erfassung, Verarbeitung, Speicherung und Übertragung von Informationen zu. In Verbindung mit Prozessen wird dann auch von Process Aware Information System (PAIS) gesprochen, auf das in Kapitel 2.2 im Detail eingegangen wird. Der Umgang mit Wissen im Sinne vernetzter Information (siehe 2.1) ist dabei etwas komplexer. Mit den Formalisierungsansätzen, auch in Folge der „Semantic-Web“ Initiative [13], wurden in den letzten 20 Jahren jedoch leistungsfähige Standards wie RDF, RDFS und OWL [30] verabschiedet, auf die in Kapitel 2.7 im Detail eingegangen wird. Diese Methodiken zur Repräsentation von Wissen können zusammen mit Ansätzen zur Konzeptionalisierung generischen Wissens und den Möglichkeiten zur Schlussfolgerung von Zusammenhängen auf Grundlage der Prädikatenlogik unter dem Begriff *Semantische Technologien* zusammengefasst werden. Dabei wurde die grundsätzliche Eignung dieser Semantischen Technologien im Kontext von Geschäftsprozessen mit unterschiedlichen Anwendungsmöglichkeiten in [30] aufgegriffen und beschrieben.

Für die unmittelbare Verbindung zwischen den beiden Forschungsbereichen zu Geschäftsprozessen und Semantischen Technologien wird die Bezeichnung „Semantische Prozessmodellierung“ [106] verwendet. In diesem Forschungsgebiet geht es grundlegend zunächst um die Eliminierung von Ambiguität in Geschäftsprozessmodellen [73]. Im Hinblick auf den Nutzen eines formalisierten statt eines semi-formalen Prozessmodells gibt es viele Beiträge, welche im Rahmen einzelner Forschungsarbeiten teilweise adressiert wurden. Mit SEMPA [52] wurde beispielsweise die semantische Beschreibung mit dem Ziel einer automatisierten Planung von Prozessmodellen verwendet. Die Arbeiten von [16, 37] sehen den Nutzen einer semantischen Prozessmodellierung vor allem in der Unterstützung des Prozessdesigners während der Modellierungsphase. In Kapitel 2.8 wird nochmals im Detail darauf eingegangen, jedoch kann vorgegriffen werden mit der Aussage, dass eine Forschungslücke besteht hinsichtlich einer Ausrichtung auf KiPs. Zudem ist keine Vorarbeit bekannt, in welcher der Nutzen eines formalisierten Prozessmodells innerhalb der Prozessausführung angestrebt wird.

Im Rahmen dieser Arbeit wird entsprechend der ausgewiesenen Forschungslücken die Entwicklung eines neuen Ansatzes verfolgt, der gezielt die Fähigkeiten semantischer Technologien dazu ausnutzt, um wissensintensive Prozesse modellieren, visualisieren und ausführen zu können. Zur

Evaluation des Ansatzes wird die Domäne von BOS-Leitstellen (Behörden und Organisationen mit Sicherheitsaufgaben) gewählt, denen Aufgaben in der Gefahrenabwehr beim Brandschutz, Rettungsdienst und Katastrophenschutz sowie polizeiliche Sicherheitsaufgaben zukommen. Dieser Domäne kann eine hohe gesellschaftliche Relevanz zugesprochen werden, gerade im Hinblick auf aktuelle Ereignisse in 2021 wie einer Pandemie, Hochwasser- oder Großschadensereignissen (Explosion in einem Chemiewerk). Zur Bewältigung solcher Ereignisse können Datentechnologien und Methoden der künstlichen Intelligenz [90] dazu beitragen, schnell richtige Entscheidungen zu treffen und so jeweilige Krisensituationen beherrschbarer zu machen. Nachfolgend wird in Kapitel 4 die Notrufabfrage als ein zentraler Prozess im alltäglichen Ablauf der Leitstellen eingeführt und aufgezeigt, dass es sich dabei um ein repräsentatives Beispiel eines KiPs handelt.

1.2 Ziele der Dissertation

Durch die Ausrichtung dieser Arbeit auf KiPs stehen deren Besonderheiten, Charakteristiken und Anforderungen an einen geeigneten Prozessmodellierungsansatz im Vordergrund. Hierauf wird gezielt in Kapitel 2.6 eingegangen und mit Bezug auf Di Ciccio et al. [32] werden acht zentrale Charakteristiken von KiPs beschrieben. Demnach sind KiPs *wissensgetrieben, kollaborationsorientiert, unvorhersehbar, sich entwickelnd, zielgetrieben, ereignisgetrieben, regelgetrieben, nicht wiederholbar*. Nachfolgend überführen Di Ciccio et al. diese Charakteristiken in 25 sehr detaillierte Anforderungen, die sich im Hinblick auf weitere Vorarbeiten [43, 50, 110, 111] zusammenfassen und für diese Arbeit in drei Kernanforderungen an einen Prozessmodellierungsansatz überführen lassen.

Diese Kernanforderungen sind *Flexibilität, Adaptivität* und *Zielorientierung*. Im Vorgriff auf Kapitel 3.1 werden diese Kernanforderungen wie folgt definiert:

1. *Flexibilität* wird als die Fähigkeit verstanden, mit vorhersehbaren sowie unvorhersehbaren Veränderungen innerhalb eines Geschäftsprozesses umzugehen.
2. *Adaptivität* wird als die Fähigkeit verstanden, einen Prozess bzw. dessen Prozessdarstellung an sich entwickelnde und verändernde Prozesszustände sowie an die individuellen Anforderungen mitwirkender Prozessbeteiligte anzupassen.

3. *Zielorientierung* wird als die Fähigkeit verstanden, Veränderungen von Prozesszielen während der Prozessausführung zu berücksichtigen und die Anwender:innen bei der Erreichung dieser Ziele zu unterstützen.

Mit Einführung der zu dieser Arbeit relevanten Grundlagen in Kapitel 2 werden in 2.3.2 einige Grundprinzipien der Prozessmodellierung detailliert eingeführt. Dabei wird deutlich, dass sich ein deklaratives Modellierungsprinzip im Vergleich zu einem imperativen Ansatz besonders hinsichtlich einer flexiblen Prozessausführung auszeichnet. Ähnlich stellt es sich im direkten Vergleich zwischen einem kontrollflussorientierten und einem datenorientierten Ansatz dar. Der datenorientierte Ansatz eignet sich besonders, um den engen Zusammenhang zwischen Daten und deren Einfluss auf die Prozessausführung eines KiPs abzubilden. Mit Blick auf eine semantische Prozessmodellierung kann noch ein weiterer Ansatz betrachtet werden. So bietet eine formale Prozessrepräsentation deutliche Vorteile gegenüber einer semiformalen Prozessbeschreibung, insofern damit ein für Mensch und Maschine gleichermaßen eindeutiges wie verständliches Prozessmodell ausgedrückt werden kann. Insbesondere die Verständlichkeit für die Maschine kann in diesem Zusammenhang dazu ausgenutzt werden einen auf Inferenzierung basierenden Prozessbeitrag zu leisten und so den Mensch in der Prozessausführung zu unterstützen.

Diese bislang noch abstrakten Möglichkeiten und Vorteile der genannten Grundprinzipien werden insbesondere in Kapitel 3 im Detail vorgestellt. Die grundsätzliche Strategie im Rahmen dieser Arbeit ist es, die hinsichtlich der Kernanforderungen als geeignet erscheinenden Ansätze in einem einheitlichen, gemeinsamen Ansatz zu kombinieren. Dies wird auch mit der ersten Forschungsfrage zum Ausdruck gebracht.

Forschungsfrage 1 (FF1)

Kann aus den drei dargelegten Grundprinzipien (einer deklarativen, datenorientierten und semantischen Prozessmodellierung) ein einheitlicher Ansatz entwickelt werden, der die Kernanforderungen wissensintensiver Prozesse hinsichtlich Flexibilität, Adaptivität und Zielorientierung erfüllt?

Die Forschungsfrage 1 adressiert somit einen generischen und domänen-unabhängigen Ansatz zur Modellierung und Ausführung von KiPs. Mit Blick auf die Definition von KiPs gemäß Gronau und Weber [43] ergibt sich noch eine weitere vielversprechende Möglichkeit. Darin heißt es - zur Erinnerung: „Ein Prozess ist wissensintensiv, wenn sein Wert nur durch die Erfüllung der Wissensanforderungen der Prozessteilnehmer geschaffen werden kann.“ Betrachtet man in dieser Definition Prozessteilnehmer als menschliche und nicht-menschliche Mitwirkende wird deutlich, dass jedwede Wissensquelle in Frage kommt, um einen Mehrwert in einem Prozess zu erwirken. In diesem Kontext wird von domänenspezifischem Wissen ausgegangen, welches in einen auf eine Aufgabenstellung ausgerichteten Prozess eingebracht wird. Diesbezüglich wäre es im Kontext eines semantischen Prozessmodells auch denkbar, dass verfügbares Wissen teilweise in Form einer ontologischen Repräsentation in eine Prozessausführung eingebunden wird. Diese Möglichkeit wird in mehreren Arbeiten aufgegriffen und im Kontext Semantischer Technologien [30] und Semantischer Prozessmodellierung [106] als ein erstrebenswertes Ziel mit einem potentiell signifikanten Mehrwert beschrieben. Auf diesem Gedanken aufbauend könnte konzeptionalisiertes, domänenspezifisches Expertenwissen mit Hilfe von Schlussfolgerungen einen expliziten Beitrag leisten, womit nicht nur durch den Menschen, sondern auch durch die KI-Methode des ontologischen Schließens ein Prozessbeitrag ermöglicht würde. Die zweite Forschungsfrage greift diese Möglichkeit auf.

Forschungsfrage 2 (FF2)

Kann in dem einheitlichen Ansatz ontologisches Wissen so integriert werden, dass dieses in einer Prozessausführung einen Beitrag leistet?

Die Forschungsfrage 2 ergänzt somit den generischen Ansatz um eine Ausrichtung auf ein konkretes Anwendungsszenario innerhalb einer Domäne. Nach der Identifikation der zentralen Forschungsfragen werden diese im Kontext der nachfolgend vorgestellten Forschungsmethodik adressiert.

1.3 Forschungsmethodik

Innerhalb der Wirtschaftsinformatik haben sich vor allem zwei Forschungsmethoden etabliert. Der erklärungs- oder verhaltensorientierte Forschungsansatz (engl. Behaviorism Research Methodology) (BRM) widmet sich dem Beobachten von bestehenden Informationssystemen, wobei die Beziehung zwischen Ursache und Wirkung sowie der Auswirkung auf das Umfeld untersucht wird. Ein designorientierter Forschungsansatz (engl. Design Science Research Methodology) (DSRM) beginnt mit der Definition eines angestrebten Ergebnisses und bietet dazu die Methodik, um sich entlang von Evaluations- und Iterationsschritten dem definierten Ziel zu nähern. Diese Arbeit orientiert sich an den DSRM Prinzipien [118, 83], um zu den Forschungsfragen geeignete Antworten zu finden und deren Wirksamkeit zu beurteilen.

DSRM unterstützt ein pragmatisches Forschungsparadigma, das die Schaffung innovativer Artefakte zur Lösung von Problemen der realen Welt fordert [54]. Im Zentrum der Methodik steht die Kreation von Artefakten, welche darauf abzielen, für eine Problemstellung neue, innovative Lösungen zu finden oder eine Verbesserung in Form einer effektiveren bzw. effizienteren Lösung zu einem bekannten Problem zu erwirken. Die Erzeugung dieser Design-Artefakte unterliegt existierenden Kerntheorien, die anhand der Kreativität, Erfahrung, Intuition und Problemlösungsfähigkeit der Forschenden angewendet, getestet, modifiziert und erweitert werden [41, 118]. Design Science ist somit eine Methode, die es ermöglicht, auf Basis von wissenschaftlichen und praktischen Kriterien ein Artefakt zu erstellen.

Hevner [55] differenziert innerhalb der Design Science Methodik drei Zyklen: Im wissenschaftlichen *Rigor-Zyklus* wird das Wissen von der theoretischen Seite hergeleitet. Im praktischen *Relevance-Zyklus* wird der Stand der Technik mit einbezogen und mit dem Rigor-Zyklus verknüpft. Im dritten Zyklus erfolgt die Kombination von Theorie und Praxis, woraus ein Artefakt abgeleitet wird, das iterativ evaluiert und überarbeitet wird. Die Verbesserung eines Artefakts liefert dann einen Beitrag zu wissenschaftlichen und praktischen Erkenntnissen. Dabei können Artefakte in Form von mathematischen Modellen, Software oder auch informalen Beschreibungen dargestellt werden und in vier Kategorien eingeteilt werden: Konstrukte, Modelle, Methoden und Instanziierungen.

Entsprechend der vorgestellten Strategie liegt der Fokus dieser Arbeit in der Kombination der verschiedenen Ansätze zu einem einheitlichen *deklarativen*, *datenorientierten* und *semantisch beschriebenen* Prozessmodell. Die

Ausprägung einer solchen Charakteristik wird festgelegt durch ein Sprachkonstrukt, welches definiert wie ein Prozess grundsätzlich modelliert werden kann. Ein solches Sprachkonstrukt kann als Metamodell bezeichnet werden, was definiert ist als die Rahmen, Regeln, Einschränkungen, Modelle und Theorien, die für die Modellierung einer vordefinierten Klasse von Problemen anwendbar und nützlich sind [40, 77]. Innerhalb des Metamodells werden alle Prozesselemente und ihre möglichen Beziehungen definiert, wodurch ein deklaratives und datenorientiertes Grundprinzip geprägt werden kann. Durch die Formalisierung dieses Metamodells in Form einer Ontologie wird zudem der Zielsetzung einer semantischen Beschreibung genüge getan. Das erste Artefakt dieser Arbeit ist somit das *Metamodell*, auf das in Kapitel 3.3.1 nochmals eingegangen und welches in Kapitel 5 ausgearbeitet wird.

Die Anforderungen hinsichtlich Flexibilität, Adaptivität und Zielorientierung erfordern zudem ein dynamisches Verfahren einer Prozessausführung [31]. Hierzu sind die üblichen Verfahren prozessausführender Systeme ungeeignet, da sich dabei ein Prozesszustand lediglich durch den sequentiellen Ablauf einer kontrollflussorientierten Prozessdefinition ergibt. Stattdessen wird ein Verfahren angestrebt, bei dem der Prozesszustand mittels Schlussfolgerung der Zustände einzelner Prozesselemente ermittelt wird. Das zweite Artefakt ist dann eine Menge an Regeln, welche diese Möglichkeiten der *Inferenzierung von Prozesszuständen* gestatten. In Kapitel 3.3.2 wird hierauf nochmals eingegangen und Kapitel 6 widmet sich der detaillierten Ausarbeitung dieses Artefakts.

Mit den ersten beiden Artefakten wird die innere Methodik eines neuen Ansatzes adressiert, unberücksichtigt ist bislang jedoch die Perspektive der Anwender:innen. Verfahren nach einem kontrollflussorientierten Ansatz bieten den mitwirkenden Personen einen strikten Rahmen und damit während der Prozessdurchführung eine starke Führung oder Unterstützung an [84]. Eine solche Prozessunterstützung kann ein Prozess nach einem deklarativen Ansatz, der ein hohes Maß an Flexibilität erlaubt und dabei gleichzeitig die Prozessziele anhand der verfügbaren Daten anpasst, zunächst nicht bieten. Diesem Defizit soll begegnet werden, indem die mitwirkenden Personen durch eine adaptive Prozessdarstellung bei der Prozessausführung unterstützt werden. Teilweise kann hierfür auf den im zweiten Artefakt gewonnenen Erkenntnissen zum Prozesszustand aufgebaut werden. Dieser Prozesszustand in Kombination mit den jeweiligen Bedürfnissen der Anwender:innen können für eine adaptive und individualisierte Prozessdarstellung verwendet werden. Das dritte Artefakt ist somit eine adaptive Prozessvisua-

lisierung (aPV), worauf in Kapitel 3.3.3 nochmals eingegangen und welches in Kapitel 7 im Detail ausgearbeitet wird.

Die drei eingeführten Artefakte bilden das methodische Grundgerüst des neuen Ansatzes, dessen Qualität hinsichtlich der Erfüllung der für KiPs formulierten Kernanforderungen überprüft werden muss. Hierzu wird eine prototypische Umsetzung erforderlich, welche die in Kapitel 5, 6 und 7 entwickelten Artefakte in einem funktionstüchtigen System zusammenfasst und realisiert. Parallel hierzu wird für eine Evaluation ein repräsentatives Anwendungsszenario benötigt, welches in Kapitel 4 eingeführt wird. Das zu evaluierende System muss dann an die entsprechende Domäne und das konkrete Anwendungsbeispiel angepasst werden. Diese Umsetzung des methodischen Grundgerüsts sowie die Ausrichtung auf ein Anwendungsszenario bildet das vierte Artefakt, worauf in Kapitel 3.3.4 nochmals eingegangen und welches in Kapitel 8 im Detail ausgearbeitet wird.

Kapitel 3 führt nochmals detailliert in die Problemstellung und den Lösungsansatz ein, erläutert die zugrundeliegende Strategie, die konkrete Umsetzung durch die Artefakte und bringt dies in Beziehung zu den beiden Forschungsfragen und den bereits eingeführten Kernanforderungen. Der übergeordnete Rahmen dieser Arbeit wird dann mit Abbildung 3.1 detailliert dargestellt.

1.4 Forschungsprojekte

Die Motivation sowie der Anwendungsbezug dieser Arbeit stammen aus der Beteiligung des Autors an zwei Forschungsprojekten, welche nachfolgend kurz vorgestellt werden.

1.4.1 SEMAFLEX

SEMAFLEX¹ hatte als Forschungsprojekt die Zielsetzung einer semantischen Integration durch ein flexibles Workflow-Management-System (WfMS) und Dokument-Management-System (DMS). Im Fokus stand die Unterstützung kleiner und mittelständischer Bauunternehmen bei der Bewältigung schwer planbarer und agiler Prozessverläufe wie beispielsweise

¹SEMAFLEX wurde gefördert durch die Stiftung Rheinland-Pfalz für Innovation, Fördernummer 1158, Titel: „Integriertes semantisches Management von Prozessen und Geschäftsdokumenten zur Unterstützung flexibler Geschäftsabläufe im Mittelstand“, Laufzeit: 2015-2017

einer Baumängelbeseitigung. Die besondere Bedeutung von Wissen, das durch die Prozessteilnehmer im Verlauf einer Prozessausführung eingebracht wird, war ein zentraler Ansatzpunkt für die Entwicklung der dieser Arbeit zugrundeliegenden Konzepte und Artefakte. Das Projekt wurde im Rahmen einer Kooperation zwischen der Hochschule Trier und der Universität Trier durchgeführt und hat den wesentlichen Einfluss von Informationen innerhalb prozessrelevanter Dokumente im Kontext eines KiPs verdeutlicht.

1.4.2 SEMANAS

SEMANAS² ist ein Forschungsprojekt mit dem Ziel einer unmittelbaren semantischen Unterstützung eines Antragstellers im landwirtschaftlichen Bereich durch ein WfMS. Im Fokus steht dabei der datenzentrierte Charakter eines Antragsverfahrens, welches als ein repräsentatives Beispiel eines KiPs betrachtet werden kann, mit dem Ziel eine flexible Prozessausführung zu gestatten. Gleichzeitig sollen durch eine semantische Integration von Expertenwissen die Anwender:innen im Verlauf der Prozessausführung unterstützt werden. Der Anwendungsbezug wurde im Projektverlauf auf die Domäne von Leitstellen und der Ausführung von Notrufabläufen ausgedehnt, da dort eine vergleichbare Aufgabenstellung existiert. Aufgrund der kurzen Laufzeit einer Notrufabwicklung vereinfacht dies wesentlich eine Evaluation im Vergleich zu den teilweise Monate bis Jahre dauernden Antragsverfahren im landwirtschaftlichen Bereich. Das Projekt wird an der Hochschule Trier in Zusammenarbeit mit der Universität Trier durchgeführt.

Beide Forschungsprojekte greifen die Besonderheiten wissensintensiver Prozesse auf und beschäftigen sich mit geeigneten Lösungsansätzen. Im Verlauf dieser Projekte wurde vom Autor an einer Reihe von Publikationen [92, 93, 91, 94, 46, 68, 45] mitgewirkt, welche die Grundlage für die vorliegende Dissertation bilden.

1.5 Generelles zur Dissertation

1.5.1 Die Zielgruppe

Von den Leser:innen werden die Grundkenntnisse der Informatik vorausgesetzt, sodass ein Basisvokabular zum Verständnis dieser Arbeit vorhanden

²SEMANAS wird gefördert vom Bundesministerium für Bildung und Forschung (BMBF), Fördernummer 13FH013IX6, Titel: „Semantisch unterstütztes Antrags-Assistenz-System“, Laufzeit: 2017-2021

ist. Angenommen werden insbesondere auch Grundkenntnisse zu Methoden der künstlichen Intelligenz und zu semantischen Informationssystemen, wobei besonders wichtige Aspekte im Kapitel 2 *Grundlagen* nochmals eingeführt und in den Kontext dieser Arbeit gerückt werden.

1.5.2 Geschlechterneutrale Formulierungen

Im Einklang mit den derzeit etablierten Regeln wird im Rahmen dieser Arbeit auf eine geschlechterneutrale Schreibweise geachtet. Zugunsten einer besseren Lesbarkeit wird dabei nicht die voll ausgeschriebene Form für *die Leserin und den Leser* gewählt. Statt dessen wird die Form des Gender-Doppelpunkts gewählt, eine Schreibweise, welche inzwischen in unserer Gesellschaft und bei den meisten *Leser:innen* einer großen Verbreitung und Akzeptanz erreicht hat. Neben der Inklusion nichtbinärer Geschlechtsidentitäten ist diese Schreibweise auch bei Verwendung einer Sprachausgabe (z.B. bei Verwendung eines ScreenReaders) aufgrund einer kleinen eingefügten Sprechpause sehr vorteilhaft.

Da diese Arbeit eine Lösung anstrebt, welche für *Anwender:innen* konzipiert wird, wird neben dieser genderneutralen Schreibweise vereinzelt auch die Ersatzform *Nutzende* verwendet.

1.5.3 Hervorhebungen

Wenn möglich wird ein Begriff oder ein Ausdruck hervorgehoben, welcher den wesentlichen Inhalt in einem Abschnitt widerspiegelt. Solche *Hervorhebungen* werden auch auf Bezeichner angewendet, die mit einer klaren und eindeutigen Bedeutung hergeleitet werden. Die gleiche *Hervorhebung* kennzeichnet ggf. auch zuvor hergeleitete Bezeichner, welche im jeweils aktuellen Kontext mit eben exakt der zuvor eingeführten Bedeutung verstanden werden sollen.

1.6 Aufbau der Dissertation

Im Folgenden werden in Kapitel 2 die Grundlagen eingeführt, welche den Ausgangspunkt der darauf aufbauenden, eigenen Beiträge darstellen. Im Wesentlichen werden hierzu auf die Vorarbeiten im Bereich des Business Process Management (BPM) zurückgegriffen, die Grundlagen Semantischer Technologien eingeführt sowie an bestehende Arbeiten zur Prozessvisualisierung angeknüpft. In diesem Zuge werden auch Forschungslücken benannt,

an die in Kapitel 3 nochmals angeknüpft wird, um die Problemstellung und den eigenen Lösungsansatz zu spezifizieren. Dabei wird der übergeordnete Rahmen dieser Arbeit verdeutlicht, der die Strategie zur Kombination der Grundprinzipien aufgreift und den Zusammenhang mit den Artefakten und den spezifizierten Kernanforderungen adressiert.

Kapitel 4 stellt dann ein repräsentatives Anwendungsszenario aus dem Bereich der Leitstellen vor. Dabei wird ein etablierter Ablauf einer Notrufabfrage eingeführt, dessen Charakter eines wissensintensiven Prozesses verdeutlicht und bestehende Defizite des State-of-Art identifiziert.

Mit Kapitel 5 wird sich der Konzeption des ersten Artefakts, dem Metamodell, gewidmet. Darin wird verdeutlicht, wie durch das Metamodell ein deklarativer und datenorientierter BPM-Ansatz ausgeprägt wird.

In Kapitel 6 wird darauf aufbauend das zweite Artefakt adressiert. Mit der Einführung von 20 Regeln wird eine Inferenzierungsmethodik definiert, welche dazu verwendet werden kann, einen Prozess entsprechend der drei kombinierten Grundprinzipien auszuführen.

Das dritte Artefakt einer adaptiven Prozessvisualisierung wird in Kapitel 7 behandelt. Auch hier werden die vorherigen Artefakte als Ausgangspunkt verwendet, um darauf aufbauend einen eigenen Lösungsansatz zu entwickeln. Hierzu werden grundlegende Verfahren zur Adaption einer Prozessdarstellung eingeführt, eine Modellierung durch Erweiterungen der aufgebauten Ontologie ermöglicht und anhand eines Beispiels wird die ausgearbeitete aPV veranschaulicht.

Im Hinblick auf eine nachfolgende Evaluation wird in Kapitel 8 die durch die ersten drei Artefakte definierte Methodik als ein viertes Artefakt in einen Systemaufbau für eine prototypische Umsetzung überführt. Dies umfasst neben der grundlegenden Umsetzung des Verfahrens auch eine domainspezifische Erweiterung, welche an das in Kapitel 4 eingeführte Anwendungsszenario anknüpft.

Die eigentliche Evaluation des neuen Ansatzes erfolgt dann in Kapitel 9, wozu zunächst die Erwartungshaltung als Hypothesen formuliert werden. Im Anschluss wird der experimentelle Aufbau und die Durchführung der Evaluation mit Disponent:innen der Leitstelle Ludwigshafen beschrieben, bevor im Anschluss die Ergebnisse der Evaluation zur Überprüfung der Hypothesen verwendet werden.

In Kapitel 10 werden dann die im Rahmen dieser Arbeit gewonnenen Erkenntnisse zusammengefasst, noch offene Fragestellungen herausgearbeitet und ein Ausblick auf anknüpfende Arbeiten und Projekte geboten.

2 Grundlagen

Mit diesem Kapitel werden die Grundlagen eingeführt, auf Basis derer die vorliegende Arbeit durchgeführt wurde. Hierzu werden Begriffe, Konzepte und Methoden etabliert und die jeweilige Definition eingeführt.

Nachfolgend wird zunächst auf grundlegende Begriffe (2.1) und ihre Bedeutung im Kontext dieser Arbeit eingegangen. Anschließend erfolgt eine Einführung *Prozessorientierter Informationssysteme* (2.2), insbesondere mit dem Schwerpunkt auf die Ausprägung von Workflowmanagementsystemen. In diesem Kontext werden die Terminologie zu *Geschäftsprozessen* (2.3) und dazu grundlegende Prinzipien eingeführt. Daran wird mit der Beschreibung etablierter *Prozessmodellierungssprachen* (2.4) angeknüpft. Im Rahmen dieser Arbeit werden einige *Charakteristiken von Geschäftsprozessen* schwerpunktmäßig betrachtet, weshalb in (2.5) auf Grundlage von Vorarbeiten hierfür eindeutige Definitionen eingeführt werden. Der Bezug zu den im Titel bereits benannten *wissensintensiven Prozessen*, deren Definition, Anforderungen und Charakteristiken werden in (2.6) beschrieben. In (2.7) werden die Grundlagen zu den in dieser Arbeit fokussierten *Semantischen Technologien* gelegt und in (2.8) mit der Verwendung im Kontext einer *Semantischen Prozessmodellierung* verknüpft. Letztlich werden in (2.9) noch Vorarbeiten hinsichtlich einer adaptiven Prozessvisualisierung eingeführt.

Innerhalb der in diesem Kapitel eingeführten und für diese Arbeit relevanten wissenschaftlichen Vorarbeiten werden auch einzelne Forschungslücken identifiziert und definiert. Diese Forschungslücken werden im nachfolgenden Kapitel aufgegriffen und zur Motivation der Forschungsfragen herangezogen.

2.1 Daten - Informationen - Wissen

Grundlegend für die nachfolgenden Abschnitte ist zunächst eine klare Differenzierung der Begrifflichkeiten *Daten*, *Information* und *Wissen*, bevor im weiteren Verlauf die darauf aufbauenden Methoden, Verfahren und Technologien eingeführt werden können. Da im öffentlichen Diskurs „die Nutzung

von Informations- und Kommunikationstechniken als Datenverarbeitung, Informationsverarbeitung und oft auch als Wissensverarbeitung bezeichnet wird, legt dies nahe, Daten, Information und Wissen seien das gleiche" [65]. Dies ist sicher nicht der Fall, wenngleich die Begriffe aus unterschiedlichen Perspektiven betrachtet auch zu teils unterschiedlichen Definitionen führen. Bei einer eher allgemeinsprachlichen Verwendung dieser Begriffe wird der gegenwarts- und praxisorientierte Bezug fokussiert, während aus der Perspektive der Nachrichtentheorie eher statistische Merkmale betrachtet werden [65]. Grundlegend für diese Arbeit ist jedoch die nachfolgend vorgestellte semiotische und betriebswirtschaftliche Betrachtungsweise.

Entsprechend der von North 2003 eingeführten Wissenstreppe [78, S. 31] sind *Daten* das Ergebnis von Zeichen, welche um eine Syntax ergänzt werden. Ergänzt man Daten um eine Bedeutung (eine Semantik), erhält man auf der nächsten Stufe *Informationen*. Vernetzt man Informationen miteinander und ergänzt diese noch um einen eindeutigen Kontext, so spricht man von *Wissen*.

Mit etwas anderen Begrifflichkeiten aber mit einer grundlegend vergleichbaren Bedeutung wird dies auch im Rahmen der Semiotik, der allgemeinen Lehre von Zeichen und Zeichenreihen, betrachtet [65]. Die darin definierte *Syntaktik* kann man dabei auf die Stufe der Daten stellen. Mit Blick auf die Wissenstreppe nach North kann nun die *Sigmatik* als eine Zwischenstufe zur Information verstanden werden, insofern die Syntaktik zunächst nur mit dem jeweils bezeichneten Objekt in Beziehung gesetzt wird. Ergänzt man dies um die konkrete Bedeutung innerhalb eines spezifischen Kontextes (wie einer Domäne), erhält man die *Semantik* und steht damit nach North auf der Stufe der Information. Letztlich wird mit *Pragmatik* noch ein Begriff geprägt, der die Semantik um die Intension des Senders einer Information ergänzt.

Dem Begriff Information kommt dabei eine zentrale Bedeutung zu, insofern es für Daten mit einer Bedeutung (Semantik) steht und gleichzeitig die Grundlage für Wissen als das Ergebnis vernetzter Informationen bildet. Bei der Verwaltung und Verarbeitung von Informationen wird dann häufig von einem *Informationssystem* gesprochen. Grundlegend besteht ein Informationssystem aus Menschen und Maschinen, die Information erzeugen und/oder benutzen und die durch Kommunikationsbeziehungen miteinander verbunden sind [39]. Im engeren Sinne wird ein Informationssystem i.d.R. als ein computergestütztes Anwendungssystem verstanden, d.h. einem Softwaresystem zur Ausführung betrieblicher Aufgaben [39].

2.2 Prozessorientierte Informationssysteme

Innerhalb der letzten Jahrzehnte durchlief die Entwicklung und Verbreitung von Informationssystemen aufgrund der gesellschaftlichen und technologischen Entwicklung einen weitreichenden Wandel. Im Zeitraum der 1970er bis zu den 1990er Jahren kam Informationssystemen vorwiegend die Aufgabe der Datenspeicherung und Verwaltung auf Grundlage von Datenmodellen und Datenbanken zu [33, S. XIII]. Mit dem Wandel vom Industriezeitalter hin zum Informationszeitalter [26, S. 1-4] veränderten sich auch die Anforderungen an die Unternehmen und verlangten nach kontinuierlichen Anpassungen der etablierten Abläufe [82, S. 1-6]. Ausgangspunkt sind dabei die Kunden mit zunehmend individualisierten Erwartungen an die jeweils angebotenen Produkte und Dienstleistungen und somit auch an die damit im Zusammenhang stehenden Prozesse, die Mitarbeiter als Prozessteilnehmer, die verwendeten Technologien sowie die erforderlichen Informationen. Alters [6] beschreibt mit diesen sechs Konzepten den grundlegenden Rahmen eines Informationssystems, worauf aufbauend von Dumas et. al [33] der Begriff PAIS, was für ein Prozessorientiertes Informationssystem steht, geprägt und definiert wurde.

Definition PAIS

A process-aware information system (PAIS) is „a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models.”
[33, p. 7]

Auf Grundlage dieser Definition fallen auch die spezialisierten und weit verbreiteten Unternehmensanwendungen wie ERP-Systeme (Enterprise Resource Planning), CRM-Systeme (Customer Relationship Management) und generelle alle Groupware-Anwendungen in die Gruppe der Prozessorientierten Informationssysteme. Diesen Anwendungen ist gemein, dass sie das Informationsmanagement mit der Ausführung und dem Management von spezifischen Unternehmensabläufen verbinden. Ebenso gilt diese Definition für ein universell einsetzbares Werkzeug wie ein WfMS. Abbildung 2.1 stellt die verschiedenen PAIS-Anwendungsvariationen dar und segmentiert diese in zwei Dimensionen.

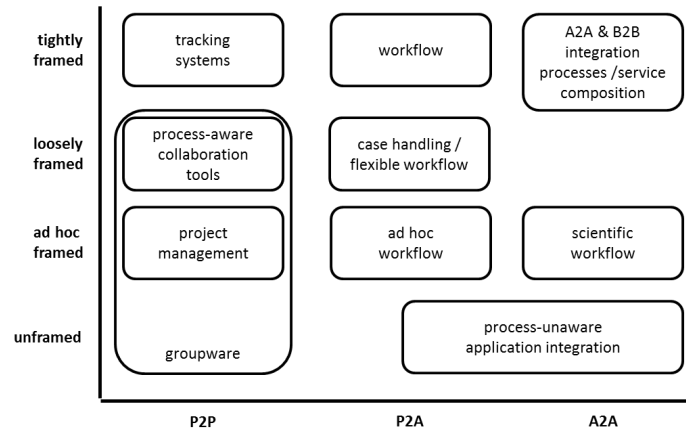


Abbildung 2.1: Varianten von PAIS nach Dumas et al. [33, S. 15]

In der horizontalen Dimension wird hinsichtlich der beteiligten Akteure (P2P - Mensch-zu-Mensch / P2A - Mensch-zu-Anwendung / A2A - Anwendung-zu-Anwendung) unterschieden. Unter P2P werden Anwendungen gruppiert, deren Fokus auf der kooperativen Zusammenarbeit zwischen Menschen liegt. Bei der P2A-Gruppe steht das Zusammenwirken von Personen und Anwendungen im Vordergrund. Unter A2A fallen Anwendungen, welche überwiegend ohne menschliche Mitwirkung durchgeführt werden.

In der vertikalen Dimension wird differenziert nach der Vorhersagbarkeit des vom jeweiligen PAIS fokussierten Prozesses. Dies erstreckt sich von ungeplanten Abläufen (unframed) über wenig geplante Prozesse (ad hoc framed und loosely framed) bis hin zu strikt vordefinierten Prozessen (tightly framed). Im Rahmen dieser Arbeit werden flexible Workflows fokussiert, also Geschäftsprozesse, welche sowohl die Zusammenarbeit zwischen Menschen und Anwendungen koordinieren und dabei gleichzeitig nur innerhalb eines lose geplanten Rahmen durchgeführt werden.

2.3 Geschäftsprozesse

Innerhalb der Wirtschaftsinformatik lassen sich zum Begriff Geschäftsprozess eine Vielzahl verschiedener Definitionen finden, welche jeweils verschiedene Aspekte und Komponenten besonders hervorheben. Im neuen Lexikon der Betriebswirtschaftslehre definiert Häberle den Begriff im Kontext von Workflow-Management wie folgt:

Definition Geschäftsprozess

Ein Geschäftsprozess ist eine Folge von Aktivitäten, die auf die Erreichung eines betriebswirtschaftlichen Ziels ausgerichtet sind. Eine Aktivität oder auch Aufgabe kann in diesem Zusammenhang als *eine betriebliche Funktion mit bestimmbarem Ergebnis* definiert werden. Ein Geschäftsprozess hat einen definierten Anfang, einen organisierten Ablauf und ein definiertes Ende. Allgemein sind Geschäftsprozesse organisationsweite oder auch organisationsübergreifende, arbeitsteilige Abläufe, in denen die anfallenden Tätigkeiten von Personen bzw. Software-Systemen koordiniert werden. [49, S. 486]

Mit Bezug auf diese Festlegung definiert Häberle weiter:

Definition Workflow

Als Workflow wird die technische Beschreibung eines Geschäftsprozesses bezeichnet. [49, S. 486]

Etwas ausführlicher definiert die Workflow Management Coalition (WfMC) den Begriff als:

Definition Workflow nach WfMC

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. [49, S. 486]

Mit Blick auf die letzte, durch die WfMC formulierte, Definition lässt sich erkennen, dass bei Geschäftsprozessen vorwiegend an einen prozeduralen Ablauf gedacht wird. Auf die unterschiedlichen Prinzipien, nach denen ein Prozess modelliert sein kann, wird im Folgenden nochmals detailliert eingegangen.

2.3.1 Terminologie

Aufbauend auf diesen grundlegenden Definitionen richtet sich der Blick auf die Entwicklung und den generellen Umgang mit Geschäftsprozessen. Diese Aspekte werden unter dem Begriff Geschäftsprozessmanagement zusammengefasst, welchen Weske definiert als:

Definition Geschäftsprozessmanagement

BPM includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes. [121, S. 5]

Demnach adressiert BPM den gesamten Lebenszyklus eines Geschäftsprozesses, ausgehend von dem initialen Design, der Administration und Konfiguration, dem Inkraftsetzen und damit Nutzen bis hin zur Bewertung. Diesen Lebenszyklus unterteilt Weske in vier Phasen [121, S. 11], welche alle miteinander in Verbindung stehen. In Anlehnung an diesen Lebenszyklus zeigt Abbildung 2.2 die einzelnen Phasen mit ins Deutsche übersetzten Bezeichnungen und definiert gleichzeitig die damit in Zusammenhang stehenden und nachfolgend verwendeten Begrifflichkeiten.

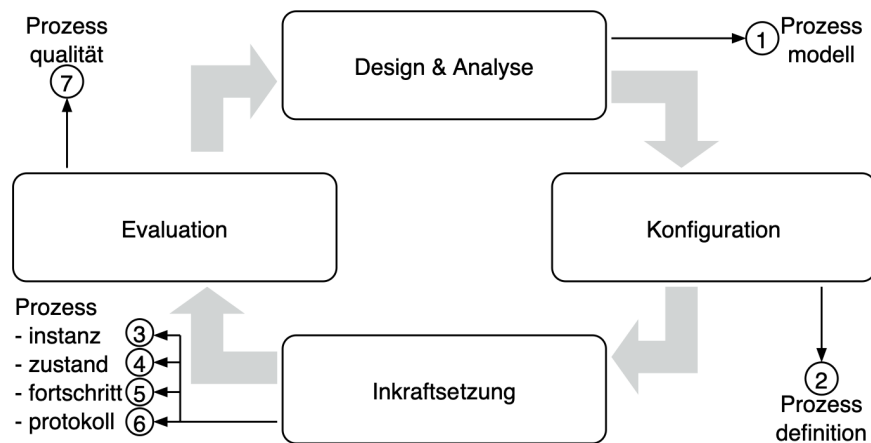


Abbildung 2.2: Lebenszyklus eines Geschäftsprozesses

Design & Analyse

Ausgangspunkt in der Design- und Analysephase ist ein Informationsmodell, welches definiert wird als eine abstrakte Repräsentation eines betriebswirtschaftlich relevanten Sachverhalts für Zwecke der Anwendungssystem- und Organisationsgestaltung [11]. *Geschäftsprozessmodelle* (nachfolgend verkürzt als *Prozessmodell* ① bezeichnet) sind spezielle Informationsmodelle, die der Repräsentation von Geschäftsprozessen dienen [12]. Entsprechend werden sämtliche Aktivitäten der Konstruktion, Anwendung und Wartung von Geschäftsprozessmodellen im Rahmen des Prozessmanagements als *Geschäftsprozessmodellierung* bezeichnet. Zumeist wird ein Prozessmodell in mehreren Iterationen überarbeitet, während stetig validiert wird, in welchem Maße das Modell das Erreichen des angestrebten Geschäftsziel unterstützt.

Ein Prozessmodell wird gebildet durch *Prozesselemente*, die sich zumeist einer der Kategorien (Aktivitäten, Dokumente, Datenobjekte) zuordnen lassen und miteinander in Beziehungen stehen. In der Literatur finden sich leicht unterschiedliche Definitionen oder Verwendungen für die Bezeichner der Kategorien. Die nachfolgende Beschreibung repräsentiert die für diese Arbeit gültige Definition des jeweiligen Begriffes.

1. *Aktivitäten* beschreiben die Tätigkeiten von Menschen, Maschinen oder Diensten, welche innerhalb eines Prozessmodells beschrieben und mit anderen Prozesselementen in Beziehung gesetzt werden.
2. *Dokumente* dienen dem Austausch von Informationen zwischen internen und externen Beteiligten. Ihrem Inhalt oder ihrer Existenz kommt innerhalb eines Prozessmodells eine Bedeutung zu, woraufhin diese ebenfalls mit anderen Prozesselementen in Beziehung gesetzt werden können.
3. *Datenobjekte* tragen eine prozessrelevante Information und sind eine digitale Repräsentation eines Objektes aus der realen Welt oder kennzeichnen einen prozessrelevanten Status.

Etabliert sind auch noch weitere Kategorien von Prozesselementen wie beispielsweise Ereignisse. Da diese im Rahmen dieser Arbeit nicht gesondert betrachtet werden, wird auf die Einführung dieser und weiterer Prozesselementkategorien verzichtet.

Konfiguration

In der Phase der *Konfiguration* wird das entworfene Prozessmodell implementiert, damit es von einem Softwaresystem ausgeführt werden kann. Hier-

zu muss das Modell um die erforderlichen Spezifikationen erweitert werden, um notwendige Dienste integrieren, geeignete Schnittstellen für Prozessbeteiligte bereitstellen sowie erforderliche Datentransaktionen verwalten und durchführen zu können [121, 13]. Das Ergebnis dieser vor allem technischen Realisierung des Prozessmodells ist eine Musterlösung, welche auf die Erreichung des initial adressierten betriebswirtschaftlichen Ziels ausgerichtet ist. Im Sinne der Definition kann es somit als eine Workflow-Definition oder auch vereinfacht als eine *Prozessdefinition* ② bezeichnet werden.

Inkraftsetzung

In der dritten Phase wird das entworfene Prozessmodell, das durch eine Prozessdefinition technisch repräsentiert wird, in einer Prozessinstanz ③ manifestiert. Eine Prozessinstanz stellt einen einzelnen konkreten Geschäftsfall dar, dessen Prozessausführung von einem Softwaresystem gesteuert wird. Hierzu werden ausführbare Aktivitäten mit den im Prozessmodell definierten Ressourcen ausgestattet und eine Ausführung wird initiiert. Durch ausgeführte Aktivitäten und generell mit jeder in die Prozessinstanz eingebrachten Aktualisierung, wie der Einwirkung von Ereignissen und bekanntgewordenen Daten, verändert sich der Status und Inhalt von Prozesselementen. Mit jeder dieser Veränderung nimmt die Prozessinstanz einen neuen Prozesszustand ④ an. In dem Maße, wie ein neuer Prozesszustand eine Prozessinstanz einem angestrebten Prozessziel näher bringt, kann auch von einem Prozessfortschritt ⑤ gesprochen und dieser gegebenenfalls durch eine geeignete Metrik beziffert werden. Mit dem Erreichen der angestrebten Prozessziele terminiert in der Regel eine Prozessinstanz. Der finale Prozesszustand zusammen mit den zeitlich dokumentierten Veränderungen der Prozesselemente wird als Prozessprotokoll ⑥ bezeichnet.

Evaluation

Auf Grundlage verfügbarer Prozessprotokolle wird in der vierten Phase das Prozessmodell evaluiert, wodurch eine Aussage zur jeweils aktuellen Prozessqualität ⑦ getroffen werden kann. Die Ergebnisse können dann für Verbesserungen und Optimierungen des Prozessmodells verwendet werden, wodurch der gesamte Zyklus erneut beginnt.

2.3.2 Grundprinzipien der Prozessmodellierung

Bislang wurden die Begrifflichkeiten für ein erstes wenngleich noch sehr abstraktes Verständnis zu Geschäftsprozessen eingeführt. Dadurch ist definiert, dass sich ein Geschäftsprozess auf die Erreichung eines betriebswirtschaftlichen Ziels ausrichtet und einem grundlegenden Lebenszyklus folgt. Offen ist jedoch die Art und Weise, wie ein Geschäftsprozess definiert wird, also das der Prozessmodellierung zugrundeliegende Prinzip. Hierzu gibt es mehrere, teils entgegengesetzte und teils kombinierbare, Ansätze, welche nachfolgend eingeführt werden.

Imperativ versus Deklarativ

Die erste und vielleicht fundamentalste Ausrichtung eines Modellierungsansatzes liegt in der Art und Weise, wie Prozesselemente miteinander in Beziehung gesetzt werden können. Analog zu den bekannten Programmierparadigmen kann auch hier zunächst in eine imperative und eine deklarative Form der Prozessmodellierung unterschieden werden. Van der Aalst bezeichnet den imperativen Modellierungsansatz als „Inside-to-Outside approach“ [116]. Bei diesem Ansatz wird explizit festgelegt, welche Aktivitäten in welcher Reihenfolge und unter welchen Umständen ausgeführt werden sollen. Es wird somit das für ein Prozessmodell erlaubte Verhalten modelliert und in Form sequentieller Ausführungspfade spezifiziert.

Bei einer deklarativen Modellierung wird von einem „Outside-to-Inside approach“ [116] gesprochen. Hierbei wird zu jedem Prozesselement beschrieben, in welchen Beziehungen es zu anderen Prozesselementen steht. Eine besondere Bedeutung im Rahmen der möglichen Beziehungen kommt dabei den „Constraints“, also den Einschränkungen zu. Mit Hilfe dieser Art von Beziehungen können innerhalb eines deklarativen Modellierungsansatzes unerlaubte oder unerwünschte Prozesszustände ausgedrückt werden. Eine grundlegende Gegenüberstellung dieser beiden Ansätze wird auch von Fahland et al. [36] behandelt. Dabei wird auch eine grundsätzliche Überführbarkeit von imperativen in deklarative Prozessmodelle und vice versa aufgezeigt.

Vereinfacht kann jedoch festgehalten werden, dass ein imperativer Ansatz zu einem eher strikten Prozessmodell führt, zu dem jede mögliche Variante einer Ausführung explizit angegeben werden muss. Ein deklarativer Ansatz hingegen führt zunächst zu einem eher lose definierten Prozessmodell, bei dem jede Ausführungsvariante erlaubt ist, sofern diese nicht gegen explizit definierte Einschränkungen verstößt.

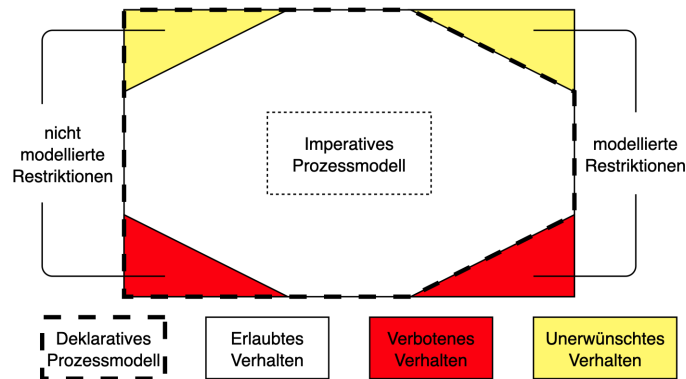


Abbildung 2.3: Prozesszustände: Imperativer versus deklarativer Modellierungsansatz in Anlehnung an [84]

Unterstrichen wird dies durch die Abbildung 2.3, welche in Anlehnung an [84] die für einen Prozess möglichen Prozesszustände mit Bezug auf das verwendete Modellierungsparadigma darstellt. Demnach kann mit einem imperativen Prozessmodell (gepunkteter Rahmen) nur eine Teilmenge der erlaubten Prozesszustände erreicht werden, abhängig von dem Umfang der explizit modellierten Ausführungspfade. Auf Basis eines deklarativen Prozessmodells kann eine Prozessinstanz grundsätzlich alle möglichen Prozesszustände annehmen, sofern diese nicht durch explizit modellierte Restriktionen eingeschränkt werden. Mit dem gestrichelten Rahmen wird ein deklaratives Prozessmodell dargestellt, welches sich teilweise durch modellierte Restriktionen vor verbotenem und unerwünschtem Verhalten abgrenzt, während andererseits durch fehlende Restriktionen manch verbotenes und unerwünschtes Verhalten bzw. entsprechende Prozesszustände noch möglich ist.

Im Ergebnis kann festgehalten werden, dass ein imperatives Prozessmodell einfach und kompakt zu definieren ist, wenn es darum geht einen eher strikten Geschäftsprozess abzubilden, der wenig Flexibilität in der Ausführung erlaubt. Jede Form von Flexibilität führt zu der Notwendigkeit, diese durch zusätzliche Ausführungspfade zu modellieren, was wiederum ein umfangreicheres Prozessmodell bedingt. Im Gegenzug ist ein deklaratives Prozessmodell einfach und kompakt, wenn keine oder nur wenige Restriktionen definiert werden müssen und somit ein großes Maß an Prozesszuständen und damit eine flexible Prozessausführung erlaubt werden soll.

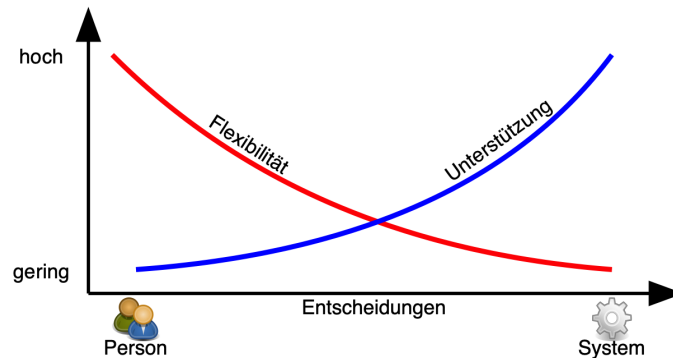


Abbildung 2.4: Flexibilität versus Unterstützung nach [84]

Darüber hinaus finden sich zum direkten Vergleich dieser beiden grundlegend unterschiedlichen Grundprinzipien weitere Arbeiten, welche beispielsweise die Verständlichkeit der jeweiligen Prozessmodelle [36] adressieren. Innerhalb der Arbeiten zu DECLARE [84] werden dabei auch die gegenläufigen Eigenschaften hinsichtlich Flexibilität und Unterstützung diskutiert und wie in Abbildung 2.4 dargestellt einander gegenübergestellt. Demnach treffen Systeme nach einem eher strikten imperativen Ansatz hinsichtlich des Prozessverlaufs die grundlegenden Entscheidungen, woraufhin diese wenig Flexibilität, jedoch ein hohes Maß an Unterstützung bieten. Im Gegensatz dazu überlassen deklarative Ansätze die Entscheidungen des Prozessverlaufs den mitwirkenden Personen, wodurch diesen ein hohes Maß an Flexibilität geboten wird, das System jedoch wenig Unterstützung bietet. Als Lösung empfehlen Pesic et al. einen flexiblen deklarativen Ansatz, der die Anwender:innen durch Empfehlungen bei der Prozessausführung unterstützt aber nicht in ihren Entscheidungen einschränkt. Dieser Herangehensweise folgt auch der im Rahmen dieser Arbeit entwickelte Ansatz, was in den nachfolgenden Kapiteln nochmals aufgegriffen wird.

Kontrollflussorientiert versus Datenorientiert

Mit dem imperativen Modellierungsansatz geht in der Regel auch ein kontrollflussorientierter Charakter einher. Hierbei stehen vor allem die Aktivitäten im Vordergrund, weshalb oft auch von einem aktivitätzentrierten Ansatz gesprochen wird. Dabei wird der Charakter vor allem durch die Verknüpfung zwischen den zumeist aus Aktivitäten bestehenden Prozess-elementen geprägt. Diese Verknüpfungen regeln vor allem den konzipierten

Prozessablauf und damit repräsentieren sie einen Kontrollfluss im Prozessmodell. Innerhalb dieser kontrollflussorientierten Perspektive eines Prozessmodells finden Daten und ihr Prozesseinfluss keinen oder nur eine geringe Berücksichtigung, weshalb nach Cohn und Hull [28] von einem *Afterthought*, einem nachträglichen Einfall, gesprochen werden sollte. Im Rahmen weiterer Forschungsarbeiten wird aufgezeigt, dass in vielen Anwendungsfällen Daten ein zentraler Prozesseinfluss zukommt, woraufhin sie neue und datenorientierte Prozessmodellierungsansätze konzipieren und im Rahmen weiterer Veröffentlichungen [59, 60] vorstellen.

Bei einem datenorientierten Prozessmodell sollen die Daten als „First-Class-Citizen“ [60] in das Prozessgeschehen eingebunden und berücksichtigt werden. Ein möglicher Ansatz der Umsetzung findet sich in ECA-Regeln (Event-Condition-Action rules) [61], welche bei definierten Ereignissen (Events), wie einem Statuswechsel bei einem Datenelement, unter Prüfung vorbestimmter Bedingungen (Conditions) festgelegte Aktionen (Actions) zur Ausführung bringen. Auf diesem Prinzip beruht auch ein Guard-Stage-Milestone (GSM) Lifecycle [60], einem ereignisgetriebenen, deklarativen und datenzentrierten BPM-Ansatz, welcher nachfolgend nochmals genauer eingeführt wird.

Semantische versus nicht semantische Prozessmodellierung

Mit der Verbreitung der inzwischen etablierten Ansätze zur Geschäftsprozessmodellierung, welche nachfolgend nochmals im Detail vorgestellt werden, hat sich auch die Verwendung graphischer Modellierungswerkzeuge verbreitet. Innerhalb der damit erzeugten graphischen Darstellung eines Prozessmodells wird in der Regel ein Teil der Bedeutung in Form von natürlich-sprachlichen Bezeichnern repräsentiert. Trägt ein Prozesselement beispielsweise ein Label wie „Bestellung prüfen“, so fügt diese Bezeichnung dem Prozessmodell eine Bedeutung bei. Bei dieser Form einer Prozessbeschreibung kann dann nur von einer semi-formalen Definition des Prozessmodells gesprochen werden, in dessen Kontext Thomas und Fellmann [106] zwei wesentliche Defizite identifizieren. Zum einen kann es zu Mehrdeutigkeiten zwischen Prozessteilnehmern führen, woraufhin sich das Modell nur eingeschränkt als Kommunikationsmedium eignet. Zum anderen kann die in den Bezeichnern repräsentierte Semantik nicht maschinell verarbeitet werden und somit beispielsweise nicht von automatisierten Systemaktivitäten berücksichtigt werden.

Im Gegensatz zu einer solch semantisch unvollständigen Prozessmodellierung werden bei einer semantischen Prozessmodellierung (SPM) alle im

Prozesskontext Bedeutung tragenden Aspekte innerhalb des Prozessmodells abgebildet. Die Basis bildet hierzu die Abbildung der grundlegenden Artefakte zur Prozessmodellierung durch eine Ontologie, eine Methodik, welche in den nachfolgenden Kapiteln nochmals im Detail eingeführt wird. Das generelle Potential einer semantischen Prozessmodellierung (SPM) wurde bereits zuvor von Hepp et al. 2005 [53] erkannt und beschrieben. Darin wird die damals noch junge Idee des Semantic Web [14] mit dem BPM verknüpft, wofür der Begriff Semantic Business Process Management (SBPM) geprägt wurde. Auch auf SPM wird nachfolgend in Abschnitt 2.8 nochmals im Detail eingegangen.

Auf Grundlage der eingeführten Kategorisierung unterschiedlicher Paradigmen werden nachfolgend einige etablierte Modellierungsansätze vorgestellt und eingeordnet.

2.4 Prozessmodellierungssprachen

Nach Fahland et al. [36, S. 441] liefert eine *Prozessmodellierungssprache* die Konzepte zur Darstellung von Prozessen. Im Verlauf der letzten Jahrzehnte haben sich eine Reihe an Modellierungssprachen etabliert, deren Ausgangspunkt überwiegend auf *Petri-Netze* zurückgeführt werden kann. Petri-Netze [27] erlauben die Beschreibung von Abläufen und nicht-deterministischen Vorgängen und eignen sich so zur Beschreibung dynamischer Systeme. Dabei repräsentiert ein Petri-Netz einen gerichteten Graphen, also ein Netz aus Knoten mit einer definierten Richtung durchführbarer Transitionen.

Aus der Initiative von Booch, Jacobsen und Rumbaugh [38] zur Konsolidierung verschiedenster objektorientierter Modellierungsansätze hat sich in den 1990ern die Unified Modeling Language (UML) entwickelt, welche 1997 von der Object Management Group (OMG) aufgegriffen und standardisiert wurde. Die darin enthaltenen Verhaltensdiagramme beschreiben die Dynamik zwischen Objekten und insbesondere Aktivitätsdiagramme beschreiben die Verhaltensaspekte eines Systems durch die Parallelisierung und Synchronisation von Aktivitäten [38].

Mit einem klaren Auftrag zur Modellierung von Geschäftsprozessen hat sich aus der UML-Initiative die inzwischen am weitesten verbreitete Prozessmodellierungssprache Business Process Model and Notation (BPMN) [5] entwickelt. Vergleichbar zur Motivation von UML wird mit BPMN eine Vereinheitlichung der verschiedenen Darstellungsformen und Modellierungssprachen in der Geschäftsprozessmodellierung verfolgt [121]. Auch hier wur-

de durch die OMG für BPMN in der Version 2.0.1 [80] ein normierter Stand eingeführt, welcher zudem als internationaler Standard ISO/IEC 19510 festgelegt wurde. Neben der graphischen Notation wurde ab Version 2.0 auch eine Ausführungssemantik festgelegt [97], was als ein Schritt hin zu einer semantischen Prozessmodellierung betrachtet werden kann. Der zentrale und auch dominierende Charakter von BPMN basiert auf den von UML bereits bekannten Aktivitätsdiagrammen und führt zu einem entsprechend imperativen, aktivitätzentrierten und kontrollflussorientierten Modellierungs- und entsprechend auch Ausführungsansatz [121].

Während BPMN als Modellierungssprache vor allem innerhalb von dezierten WfMS zum Einsatz kommt, bleiben andere spezialisierte Informationssysteme (CRM-, ERP-Systeme), wie sie innerhalb einer heterogenen Systemlandschaft eines Unternehmens verbreitet sind, oft isoliert und können kaum oder nur mit zusätzlichen Schnittstellen in Prozesse eingebunden werden [102]. Zur Lösung dieser Problematik wird eine Komposition von Web-Services durch eine gezielte Orchestrierung angestrebt. Bei Web-Services kann es sich dabei um einfache Dienste bis hin zu komplexen betrieblichen Anwendungen handeln, auf deren Funktionalität über eine standardisierte, plattformunabhängige Schnittstelle zugegriffen werden kann [102]. Darauf aufbauend definiert die W3C eine Orchestrierung wie folgt: „An orchestration defines the sequence and conditions in which one Web service invokes other Web services in order to realize some useful function. I.e., an orchestration is the pattern of interactions that a Web service agent must follow in order to achieve its goal“ [48]. Zur Beschreibung solcher Orchestrierungen standardisierte die W3C mit Web Service Description Language (WSDL) eine plattformunabhängige und protokollunabhängige Sprache. Darauf aufbauend wurde von IBM, BEA Systems und Microsoft mit der Web Service Business Process Execution Language (WS-BPEL) eine weitverbreitete Sprache zur Beschreibung von Geschäftsprozessen auf Basis einzelner Aktivitäten durch Web-Services etabliert [102]. Dieser Ansatz löst ein existierendes Problem vieler Unternehmen mit einer heterogenen Systemland und folgt dabei einem überwiegend imperativen und kontrollflussorientierten Grundprinzip, bei dem die Verknüpfung von Aktivitäten im Vordergrund stehen.

Die Ereignisgesteuerte Prozesskette (EPK) hat sich als ein weiterer Standard zur Beschreibung kontrollflussorientierter Prozessmodelle etabliert [79]. Dieser Ansatz ist zentraler Bestandteil der SAP-Referenzmodelle und bildet die Grundlage der ARIS-Konzepte von IDS Scheer. In zentraler Position finden sich bei diesem Ansatz neben Aktivitäten auch Ereignisse, welche relevante Zustandsübergänge und Bedingungen repräsentieren [79].

Die grundlegende Charakteristik der EPK ist weitgehend vergleichbar mit BPMN, insofern es sich um einen imperativen, kontrollflussorientierten Ansatz mit einer semi-formalen Prozessmodellierung handelt.

Die Grenzen dieser kontrollflussorientierten Ansätze wurden ab Anfang der 2000er im Rahmen vieler wissenschaftlicher Arbeiten [74, 28, 59] adressiert. Ins Zentrum rückte immer mehr die Bedeutung von Daten und deren Einfluss auf die Abläufe von Geschäftsprozessen, woraufhin neue Ansätze entwickelt wurden. Bei vielen Studien wurde hierzu mit dem Begriff „Business Artifacts“ ein Konzept etabliert, welches Daten und die damit in Verbindung stehenden Zustandsveränderungen und Operationen repräsentiert [28]. Ins Blickfeld geriet dabei die gesamte Lebensspanne von Datenobjekten, woraus sich beispielsweise der Guard-Stage-Milestone (GSM) Ansatz [59] entwickelte. Die GSM Lebenszyklen sind in ihrer Beschreibung wesentlich deklarativer als die Zustandsautomaten anderer Modellierungsansätze und basieren auf einer Form von Event-Condition-Action Regeln [59]. Solche Artefakt-zentrierten Ansätze wurden auch mit deklarativen Prozessmodellierungsansätzen kombiniert [110] und hinsichtlich ihrer Eignung zur Ausführung wissensintensiver Prozesse untersucht, ein Aspekt der nachfolgend nochmals im Detail erläutert wird.

Generell wurde der imperative Ansatz im Rahmen diverser Arbeiten in Frage gestellt und mit den Möglichkeiten und Vorteilen einer deklarativen Prozessmodellierung verglichen. Mit *DecSerFlow* [1] wurde die Idee der Service-Orchestrierung mit dem bei WSDL und WS-BPEL etablierten imperativen Ansatz aufgegriffen und mit einer deklarativen Serviceablaufsprache umgesetzt. Zur Ausführung lose strukturierter Prozesse wurde mit DECLARE [84] ein deklarativer Prozessmodellierungsansatz vorgestellt, der gezielt auf die Definition von Constraints (Einschränkungen) zur Vermeidung unerwünschter Prozesszustände setzt. Mit Blick auf diese und weiteren Studien wurde der Bedarf nach einem standardisierten Ansatz deutlich, der die Vorteile einer deklarativen und daten-zentrierten Prozessmodellierung zur Beschreibung schwach strukturierter Prozesse erlaubt [69]. Hierzu etablierte die OMG mit Case Management Model and Notation (CMMN) einen neuen Standard, der die starre kontrollflussorientierte Charakteristik von BPMN überwindet und eine wesentlich flexiblere Prozessausführung ermöglicht. Die Flexibilität wird dabei wesentlich durch Sentries (Wächter) ermöglicht, welche Ereignisse aber auch Zustände von Datenelementen überwachen und gezielt zur Aktivierung von Aktivitäten innerhalb einer Prozessinstanz genutzt werden können. Der Zugewinn an Flexibilität geht dabei in der Regel zu Lasten der Prozessunterstützung, ein Aspekt der von van Aalst et al. [111] gezielt aufgegriffen wurde.

2.5 Charakteristiken von Geschäftsprozessen

Im Zusammenhang mit Geschäftsprozessen werden im Rahmen wissenschaftlicher Arbeiten häufig die unterschiedlichen Charakteristiken und Merkmale von Modellierungsansätzen beschrieben und hinsichtlich ihrer jeweiligen Bedeutung für spezifische Anwendungsfälle bewertet. Dabei geht es oft um sehr grundsätzliche Aspekte wie die Ausdrucksmächtigkeit [125, 115, 112] und Vollständigkeit [43, 115] einer Modellierungssprache. Oft werden aber auch die für eine Prozessausführung wesentlichen Charakteristiken analysiert, meist auch mit Blick auf spezifische Anwendungsfälle oder einzelne Domänen [116, 75, 87]. Im Rahmen dieser Arbeit werden drei Kerncharakteristiken von Geschäftsprozessmodellen fokussiert, welche sich aus verschiedenen Arbeiten [32, 43, 94] zu KiPs motivieren. Dabei dreht es sich um die Charakteristiken der Flexibilität, Adaptivität und Zielorientierung. Ihre Bedeutung hinsichtlich KiPs wird im Abschnitt 2.6 nochmals gezielt erläutert. Nachfolgend werden die genannten Kerncharakteristiken zunächst eingeführt und eindeutig definiert.

2.5.1 Flexibilität

Mit Blick auf das breite Feld verfügbarer Forschungsarbeiten und Publikationen muss festgestellt werden, dass der Begriff Flexibilität nicht einheitlich und nicht mit einer klaren und eindeutigen Bedeutung verwendet wird. Weber und Reichert [120, S. 43] überspannen mit diesem Begriff drei große Bereiche: Flexibilität, welche durch Ausführungsoptionen innerhalb einer Prozessinstanz ermöglicht wird, Flexibilität durch die Anpassung eines Prozessmodells zur Laufzeit sowie Flexibilität durch die Evolution einer Prozessdefinition. Van Aalst definiert Flexibilität als die Fähigkeit, mit vorhersehbaren sowie unvorhersehbaren Änderungen innerhalb einer Prozessausführung umzugehen [116, S. 25]. Hierzu wurde eine Differenzierung [98] eingeführt, welche mit der in Abbildung 2.5 dargestellten Grafik illustriert wird.

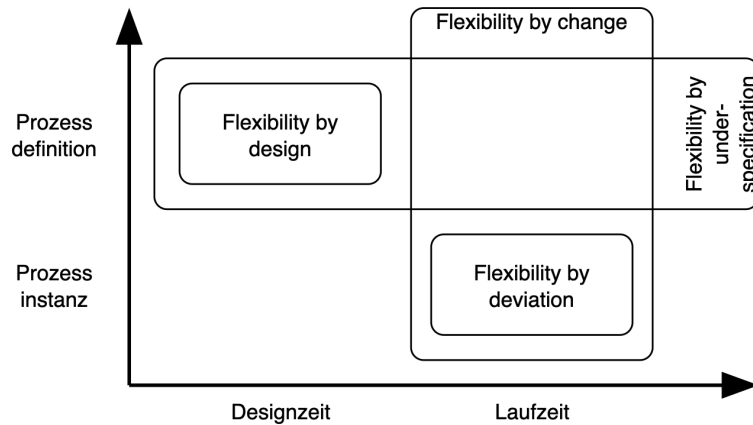


Abbildung 2.5: Differenzierung der Prozessflexibilität in Anlehnung an [116]

Unterschieden wird in der vertikalen Achse zwischen Prozessdefinition und Prozessinstanz und damit in den Ort, in dem die zur Schaffung von Flexibilität erforderlichen Anpassungen am Prozessmodell stattfinden. Über die horizontale Achse wird der Zeitpunkt differenziert, zu dem solche Anpassungen durchgeführt werden.

Flexibility by Design

„Flexibility by design“ (Synonym zu „Flexibility by definition“ [116]) bezeichnet den Fall, wenn innerhalb des Prozessmodells alle möglichen Ausführungspfade zur Designzeit innerhalb der Prozessdefinition bereits angelegt werden. Dies setzt natürlich voraus, dass alle möglichen Ausführungsvarianten bereits vor der Instanziierung einer Prozessinstanz bekannt sind. Anwender:innen können dann während der Ausführung der Prozessinstanz je nach Prozesszustand von vordefinierten Verzweigungen Gebrauch machen und so ein gewisses Maß an Flexibilität nutzen. Wird viel Flexibilität in der Ausführung benötigt, führt dies zwangsläufig zur Modellierung einer entsprechend großen Anzahl an Ausführungsvarianten, wodurch das Prozessmodell komplex und unübersichtlich wird. Dabei bleibt das Prozessmodell jedoch beschränkt auf die vorhersehbaren Änderungen einer Prozessausführung und unvorhersehbare Abweichungen werden nicht ermöglicht, weshalb der Ansatz „Flexibility by design“ als äußerst limitiert eingestuft werden muss.

Flexibility by Underspecification

Bei dem Ansatz „Flexibility by underspecification“ werden gezielt einzelne Bereiche eines Prozessmodells innerhalb der Prozessdefinition nicht ausmodelliert, sondern mit Platzhaltern ausgestattet. Nach der Instanziierung wird dann die Möglichkeit eingeräumt, innerhalb der Platzhalter jeweils unterschiedliche und zur Situation passende individuelle Aktivitäten durchzuführen. Dies erfordert jedoch die Adaption des Prozessmodells zur Laufzeit, wodurch ein Teil der Arbeit zur Erstellung einer Prozessdefinition in die Prozessinstanz verschoben wird und damit von den Anwender:innen geleistet werden muss. Bei dem sogenannten „Late binding“ [98] werden vordefinierte Prozessfragmente zur Designzeit angelegt, aus denen zur Laufzeit nur noch das passende Fragment ausgewählt werden muss. Im Unterschied dazu werden beim „Late modelling“ [98] keine Prozessfragmente vorbereitet und den Anwender:innen kommt die Aufgabe zu, den entsprechenden Platzhalter mit einem eigenen Ablauf auszumodellieren. Das Maß an Flexibilität hinsichtlich vorhersehbarer und unvorhersehbarer Änderungen ist innerhalb von Platzhaltern und beim beschriebenen „Late modelling“ grundsätzlich unbeschränkt, allerdings steigt mit jedem Zugewinn an gewonnener Flexibilität der Aufwand für die Anwender:innen, den Inhalt der Platzhalter zur Laufzeit zu definieren. Der Nachteil liegt somit in der erforderlichen Qualifikation der zur Laufzeit eingebundenen Personen, sowie einem wesentlich erhöhten Arbeitsaufwand.

Flexibility by Change

Mit dem Ansatz „Flexibility by change“ [98] wird ein grundsätzlich vollständiges Prozessmodell instanziiert und als Prozessinstanz ausgeführt. Im Falle unvorhergesehener Veränderungen im Prozessablauf besitzen die Anwender:innen die Möglichkeit, das Prozessmodell zur Laufzeit zu ändern. Wie der vorherige Ansatz basiert auch in diesem Fall die Flexibilität auf der Adaption des Prozessmodells an den spezifischen Bedarf einer einzelnen Prozessinstanz. Unterschieden wird dann jedoch in die Auswirkung solcher Adaptionen. Im einfachsten Fall wird lediglich die einzelne Prozessinstanz angepasst. Wenn der Bedarf einer Anpassung jedoch grundsätzlicher Natur ist und über eine einzelne Instanz hinaus geht, wird eine Anpassung der Prozessdefinition und gegebenenfalls anderer laufender Prozessinstanzen angestrebt. Dies zieht dann weitere Herausforderungen nach sich, wie eine konfliktfreie Anpassung laufender Prozessinstanzen oder die generelle Einhaltung von Compliance-Richtlinien und Qualitätsanforderungen. Auch in

diesem Fall geht die gewonnene Flexibilität zu Lasten der Anwender:innen, worin sich eine geringe Akzeptanz [98] begründen lässt.

Flexibility by Deviation

Bei „Flexibility by deviation“ [98] entspringt die Flexibilität nicht aus der Anpassung eines Prozessmodells, sondern aus der Fähigkeit, zur Laufzeit von einem vordefinierten Prozesspfad abweichen zu können. Diese Abweichungen beschränken sich jeweils auf eine einzelne Prozessinstanz und wirken sich nicht bis auf eine zugrundeliegende Prozessdefinition aus. Die grundsätzliche Herausforderung liegt dann in der Entscheidung, welche Abweichungen erlaubt werden können und welche gegen Compliance-Richtlinien verstoßen.

Entsprechend dieser Differenzierung verschiedener Möglichkeiten zur Schaffung von Flexibilität wird der Begriff nach der Sichtweise von van der Aalst [116, S. 25] verwendet und nachfolgend verwendet.

Definition: Flexibilität

Flexibilität ist die Fähigkeit, mit vorhersehbaren sowie unvorhersehbaren Veränderungen innerhalb eines Geschäftsprozesses umzugehen, während der grundsätzliche Aufbau eines Prozessmodells erhalten bleibt.

2.5.2 Adaptivität

Mit den Einschränkungen statischer und kontrollflussorientierter Modellierungsansätze haben sich diverse Forschungsarbeiten beschäftigt und die Notwendigkeit der Adaptivität wurde dabei zu einem zentralen Forschungsgegenstand. Unter der Begrifflichkeit Adaptivität wird dabei verstanden, dass das Prozessmodell einer Prozessdefinition oder Prozessinstanz strukturell verändert wird [8, 114]. Im vorherigen Abschnitt zur Flexibilität sind bereits Ansätze beschrieben, welche die Flexibilität über die Adaption des Prozessmodells erwirken. Dies korrespondiert mit Arbeiten [114, 113] zu Beginn der 2000er, in denen differenziert wird zwischen den Herausforderungen im Umgang mit „momentary changes“ (vorübergehende Änderungen) und „evolutionary changes“ (evolutionäre Änderungen). Im Zusammenhang damit wird auch die Adaption von Prozessinstanzen (*momentary changes*) und Prozessdefinitionen (*evolutionary changes*) diskutiert, was sich in Abbildung

2.5 in den beschriebenen Kategorien *Flexibility by change* und *Flexibility by underspecification* wiederfindet. Mit ADEPT2 [89] wird beispielsweise ein Ansatz vorgestellt, der sich gezielt mit den beschriebenen Möglichkeiten eines *Adaptiven Prozess Managements* auseinandersetzt. Basten et al. [8] platziert adaptive Workflows zwischen strikten kontrollflussorientierten Ansätzen und Adhoc-Workflows mit der Perspektive, dass sich ein guter Prozesssupport mit Flexibilität durch die Fähigkeit zur Anpassung an erforderliche Veränderungen kombinieren lassen. In [47] wird der Ansatz adaptiver Prozessmanagement-Ansätze wie CAKE [72] mit den Möglichkeiten fallbasierter Verfahren verglichen und die Stärken und Schwächen werden analysiert. Mit Blick auf diese breite Auswahl an Forschungsarbeiten kann aber eine Gemeinsamkeit identifiziert werden, welche sich in der nachfolgenden Definition widerspiegelt.

Etablierte Definition: Adaptivität

Adaptivität ist die Anpassung eines Prozessmodells mit dem Ziel Flexibilität zu ermöglichen.

Diese Definition der Begrifflichkeit Adaptivität spiegelt dabei eine technische Perspektive im Umgang mit einem Prozessmodell wider. Da die Basis dieser Arbeit jedoch ein deklaratives Prozessmodell bildet, dessen Flexibilität inhärent ist und keine Adaption im Sinne der vorgestellten Definition erfordert, wird noch eine weitere Perspektive aufgegriffen.

Eine Adaption des Prozessmodells im beschriebenen Sinne wird je nach Detailgrad einer Prozessvisualisierung den Anwender:innen gegenüber als ein veränderter Prozessgraph dargestellt. Eine solche Veränderung in der Prozessvisualisierung muss jedoch nicht zwangsläufig das Ergebnis einer Adaption im Prozessmodell sein. Es kann sich dabei auch um eine modifizierte Prozessdarstellung handeln, wie es beispielsweise in den Arbeiten von Bobrik et al. [21] vorgestellt wird. Die Anwender:innen könnten dann nicht ohne weiteres differenzieren, ob die veränderte Prozessdarstellung auf eine Adaption des Prozessmodells oder nur auf eine Adaption der Prozessvisualisierung zurückzuführen ist.

Im weiteren Verlauf dieser Arbeit wird Adaptivität als eine Anforderung für die Bewältigung der Herausforderungen im Umgang mit KiPs eingeführt. Dabei ist nicht die Anforderung an die Veränderung eines Prozessmodells zu verstehen. Vielmehr beziehen sich die Anforderungen auf die An-

wender:innen, welche eine Unterstützung durch das prozessführende System benötigen. Hierzu wird in Kapitel 7 eigens eine Methodik für eine anwenderbezogene adaptive Prozessvisualisierung vorgestellt. Im Rahmen dieser Arbeit und mit Bezug auf die noch kommenden Anforderungen von KiPs hinsichtlich Adaptivität soll statt der auf das Prozessmodell begrenzten Definition von Adaptivität eine weitergehende Definition des Begriffes genutzt werden.

Erweiterte Definition: Adaptivität

Adaptivität wird fortan als die Fähigkeit verstanden, einen Prozess bzw. dessen Prozessdarstellung an sich entwickelnde und verändernde Prozesszustände sowie an die individuellen Anforderungen mitwirkender Personen anzupassen.

2.5.3 Zielorientierung

Die Zielorientierung (engl. Goal Orientation) von Geschäftsprozessen wird in der Literatur nicht sehr häufig aufgegriffen und diskutiert, weshalb auch nicht auf ein breites Grundverständnis zu dieser Begrifflichkeit verwiesen werden kann. Hingegen wird häufig auf das angestrebte Ziel von Geschäftsprozessen als das zentrale Motiv von PAIS verwiesen. Im Zusammenhang mit flexiblen und adaptiven Prozessen beschreibt van Aalst et al. [117] ein „business goal“ aus einer anderen Perspektive. Während kontrollflussorientierte Prozesse festlegen, welche Aktivitäten durchgeführt werden sollen um ein vordefiniertes Ziel zu erreichen, wird bei einem für KiPs geeigneten Ansatz wie „Case handling“ davon gesprochen, dass Aktivitäten durchgeführt werden können, welche zur Erreichung eines „business goals“ etwas beitragen. Damit wird der Weg zu einem Ziel adressiert, das Ziel wird als statisch betrachtet.

Aus einer ähnlichen Perspektive betrachtet dies auch Marjanovic et al. [70], wobei die Bedeutung eines gemeinsamen und geteilten Ziels von Wissensarbeitern im Zusammenhang mit KiPs hervorgehoben wird.

Eine Definition der Begrifflichkeit „Goal Orientation“ findet sich bei Mundbrod et al. [76, S. 6], in der auch das zwischen Wissensarbeitern geteilte „Common Goal“ fokussiert wird, darüber hinaus aber auch die Möglichkeit von sich verändernden Zielen während eines Prozessverlaufs mit aufgegriffen wird.

Dieser Blickwinkel fließt auch in die Definition von Di Ciccio et al. [32] mit ein: „Moreover, goals may be modified or invalidated as a consequence

of occurring events, which had an impact on process state and execution context.”

In Anlehnung an diese Vorarbeiten wird für diese Arbeit folgende Definition festgelegt.

Definition: Zielorientierung

Zielorientierung beschreibt das mit den Prozessteilnehmern gemeinsame Verfolgen geteilter Prozessziele, welche a priori feststehen oder sich im Prozessverlauf verändern können.

Zielorientierung aus technischer Sicht ist gegeben, wenn ein System die Anwender:innen bei der Erreichung von Prozesszielen unterstützt und dabei neben vordefinierten Zielen auch die Veränderlichkeit von Prozesszielen während der Prozessausführung erlaubt.

2.6 Wissensintensive Prozesse

Wissensintensive Prozesse haben sich innerhalb der BPM-Domäne zu einem bedeutenden Forschungsthema entwickelt [120, 31], nicht zuletzt aufgrund der zentralen Rolle von Wissensarbeitern innerhalb von Unternehmen. Dies wird sowohl durch quantitative Werte gestützt, demnach zwischen 25-40% [23] der Arbeitskraft von Wissensarbeitern geleistet wird, als auch durch qualitative Bewertungen untermauert, wonach der Beitrag der Wissensarbeiter einen wesentlichen Einfluss auf den Erfolg eines Unternehmens hat [32]. Eine besondere Rolle spielt dabei der Austausch zwischen Wissensarbeitern und damit der kollaborative Aspekt innerhalb einer Prozessausführung, sowie die Tatsache, dass die Grundlagen von KiPs eher in „Best Practices“ als in ablauforientierten Prozessen zu finden sind [70]. Gronau und Weber [43] identifizierten als einen grundlegenden Bedarf, dass mit Blick auf KiPs ein BPM-Framework auch die Dimension von Wissen und Zusammenarbeit innerhalb des Prozess-Lebenszyklus mit abbilden muss. Damit verschiebt sich das Ziel eines entsprechenden BPM-Frameworks von der Automation und Steuerung von Prozessschritten hin zu einem System zur Entscheidungsunterstützung und zur Kollaboration zwischen Wissensarbeitern [32].

Nachfolgend wird zunächst eine eindeutige Definition von KiPs erarbeitet, woraufhin anschließend die Charakteristiken im Detail beschrieben werden.

2.6.1 Definition von KiPs

Innerhalb der Literatur findet sich keine eindeutige oder breit anerkannte Definition wissensintensiver Prozesse, jedoch kann aus den bestehenden Arbeiten ein grundlegender Konsens identifiziert werden. Dabei können KiPs zunächst grundsätzlich in den noch weitgehend unerschlossenen Bereich zwischen Geschäftsprozessmanagement und Wissensmanagement eingeordnet werden [32]. Davenport et al. fokussiert dabei vorwiegend eine durch die mitwirkenden Menschen in den Prozesskontext eingebrachte Form von Wissen und unterscheidet dabei zwischen explizitem und implizitem Wissen [29]. Während *explizites* Wissen in eine repräsentative Beschreibung überführt und dann einfach kommuniziert, verarbeitet und gespeichert werden kann, ist das Formalisieren von *implizitem* Wissen in Form von Erfahrungen und mentalen Modellen nicht einfach möglich [32]. In dem impliziten Wissen liegt die eigentliche Herausforderung im Umgang mit KiPs, was sich auch als Statement bei Gronau et al. wiederfindet: „knowledge-intensive business processes deal very much with creating and using tacit knowledge from many participants“ [42]. Mit dieser Perspektive wird dann ein Prozess auch als wissensintensiv bezeichnet, wenn „its value can only be created through the fulfillment of the knowledge requirements of the process participants“ [43].

Mit Blick auf die durch verschiedene Arbeiten hervorgehobenen kennzeichnenden Aspekte von KiPs hat Vaculin et al. [110] eine eher ganzheitliche Definition eingeführt, welche auch die Grundlage für diese Arbeit bildet.

Definition: Knowledge-intensive Processes (KiPs)

KiPs are „... processes whose conduct and execution are heavily dependent on knowledge workers performing various interconnected knowledge intensive decision making tasks. KiPs are genuinely knowledge, information and data centric and require substantial flexibility at design- and run-time“ [110]

2.6.2 Charakteristiken von KiPs

Innerhalb der bereits referenzierten sowie weiteren Forschungsarbeiten werden unterschiedliche Charakteristiken von KiPs, teilweise mit unterschiedlichen Bezeichnungen, beschrieben. Darauf aufbauend hat Di Ciccio et al. eine

vereinheitlichte Zusammenstellung der wesentlichen Merkmale [32] vorgestellt, welche als wesentliche Grundlage dieser Arbeit verwendet werden.

Nachfolgend werden die acht zentralen Charakteristiken von KiPs vorgestellt und kurz erläutert.

1. Wissensgetrieben: Der Status und die Verfügbarkeit von Daten und Wissensobjekten bestimmen die menschliche Entscheidungsfindung und beeinflussen direkt den Ablauf von Prozessaktionen und -ereignissen.
2. Kollaborationsorientiert: Die Erstellung, Verwaltung und Ausführung von Prozessen erfolgt in einer kollaborativen Mehrbenutzer-Umgebung, in der menschenbezogenes und prozessbezogenes Wissen von und zwischen Prozessbeteiligten mit unterschiedlichen Rollen gemeinsam erstellt, geteilt und übertragen wird.
3. Unvorhersehbar: Der genaue Aktivitäts-, Ereignis- und Wissensfluss hängt von situations- und kontextspezifischen Elementen ab, die möglicherweise nicht a priori bekannt sind, sich während der Prozessausführung ändern und über verschiedene Prozessfälle hinweg variieren können.
4. Sich entwickelnd: Der tatsächliche Handlungsablauf ergibt sich allmählich während der Prozessausführung und wird Schritt für Schritt festgelegt, wenn mehr Informationen zur Verfügung stehen.
5. Zielgetrieben: Der Prozess entwickelt sich durch eine Reihe von zu erreichenden Zielen, Zwischenzielen oder Meilensteinen.
6. Ereignisgetrieben: Der Prozessverlauf wird durch das Auftreten verschiedener Arten von Ereignissen beeinflusst, welche die Entscheidungsfindung der Wissensarbeiter beeinflussen.
7. Regelgetrieben: Prozessbeteiligte können von Einschränkungen und Regeln beeinflusst werden oder müssen diese einhalten, welche die Ausführung von Aktionen und die Entscheidungsfindung beeinflussen.
8. Nicht wiederholbar: Die Schritte, die zur Bearbeitung eines bestimmten Falles oder einer bestimmten Situation unternommen werden, sind kaum wiederholbar, d. h. verschiedene Ausführungen eines Prozesses weichen voneinander ab.

2.6.3 Anforderungen von KiPs

Ausgehend von diesen Charakteristiken leiten Di Ciccio et al. [32] 25 grundsätzliche Anforderungen an ein für KiPs geeignetes WfMS ab, welche in sieben Anforderungskategorien (AK) gruppiert werden. Nachfolgend werden diese Kategorien vorgestellt und die einzelnen Anforderungen (engl. requirements (R)) verkürzt beschrieben.

AK1: Daten

Die Bedeutung der Daten für KiPs wird durch vier ausformulierte Anforderungen unterstrichen. Dies betrifft zum einen den Bedarf an einem geeigneten Informationsmodell (R1), das alle im Prozess relevanten Daten und ihren Einfluss auf andere Prozesselemente abzubilden vermag. Mit Blick auf die Unvorhersehbarkeit wird zudem das Modellieren von Datenbeziehungen sowohl zur Design- als auch zur Laufzeit (R2) gefordert. Zudem wird sowohl ein technischer Zugriff auf Daten unterschiedlicher Quellen (R3) sowie der gemeinsame Zugang (R4) aus einer kollaborativen Teamperspektive als Anforderung formuliert.

AK2: Wissensgetriebene Aktivitäten

In dieser Kategorie wird die Bedeutung von Daten zur Aktivierung und zur Durchführung von Aktivitäten im Kontext von KiPs adressiert. Dies gilt wieder sowohl für den Designzeitpunkt als auch für den Ausführungszeitpunkt eines Prozesses.

AK3: Regeln und Einschränkungen

Der Einfluss von Gesetzen, Compliance-Richtlinien und generellen Unternehmensregelungen auf Prozessabläufe muss formalisiert werden können und ebenfalls innerhalb des Ausführungszeitpunktes an unvorhergesehene Aspekte angepasst werden können.

AK4: Ziele

Mit Blick auf den zielorientierten Charakter von KiPs müssen Prozessziele sowohl „a priori“ innerhalb einer Prozessdefinition modelliert werden können, als auch zur Prozesslaufzeit innerhalb einer Prozessinstanz auf Basis der sich entwickelten Datenelemente angepasst werden können.

AK5: Prozesse

Innerhalb dieser Kategorie werden sieben einzelne Anforderungen zusammengefasst, startend mit dem Bedarf nach verschiedenen Prozessmodellierungsansätzen (R11), um einem unterschiedlichen Grad an Strukturiertheit nachkommen zu können. Zur Darstellung des Prozesswissens (R12) wird eine aggregierte Perspektive auf Daten, Aktivitäten, Einschränkungen und Prozesszielen sowie auf deren Beziehungen untereinander gefordert. Die Unvorhersehbarkeit von KiPs verlangt nach einem hohen Maß an Flexibilität (R13) während der Prozessausführung für die mitwirkenden Wissensarbeiter, insbesondere hinsichtlich der Editierbarkeit von Daten sowie Ausführbarkeit von Aktivitäten. Dies spiegelt sich auch in der Forderung wider, dass KiPs in der Lage sein müssen, mit unvorhergesehenen Ereignissen und Ausnahmen (R14) umzugehen. Bei länger laufenden Prozessen muss zudem berücksichtigt werden, dass nicht nur Veränderungen innerhalb einer Prozessinstanz auftreten, sondern auch das Umfeld solchen Veränderungen unterliegen kann. Hierzu wird die Anforderung (R15) formuliert, dass laufende Prozessinstanzen auch an Änderungen in einem zugrundeliegenden Prozessmodell reagieren und diese migrieren können. Gleichzeitig lohnt ein Blick in die Vergangenheit, als dass in den Protokollierungen abgeschlossener Instanzen (Eventlogs) wertvolles Erfahrungswissen steckt. Daraus resultiert die Anforderung (R16) an Mechanismen zur Erschließung und Integration dieses Wissens in das jeweilige Prozessmodell. Diese Forderung nach Lernen wird ausgedehnt auf heterogene Datenquellen (R17), welche vergleichbar zu Eventlogs prozessrelevantes Wissen beinhalten können.

AK6: Wissensarbeiter:innen

Im Zentrum der generellen Anforderung an Kollaboration steht zunächst ein ausgedehntes Konzept (R18) zur Modellierung von Rollen und Fähigkeiten der involvierten Personen. Daran knüpft der Bedarf nach einer klaren Formalisierung der Interaktionen zwischen Wissensarbeiter:innen (R19) an. Aufgrund des datenorientierten Charakters von KiPs geht mit der in R13 geforderten Veränderbarkeit von Daten auch eine hohe Verantwortung einher, woraus sich die Anforderung nach einem umfangreichen Rechtemodell (R20) auf Basis von Rollen und einzelnen Datenelementen ableiten lässt. Mit Blick auf die sich entwickelnde Natur von KiPs Prozessinstanzen ergibt sich die Forderung, auch während der Prozessausführung weitere Wissensarbeiter:innen aufnehmen zu können (R21) sowie deren Rechte und Privilegien festlegen zu können (R22). Aufgrund der Tatsache, dass viele Prozessverän-

derungen und Prozessentwicklungen in einem direkten Zusammenhang mit Entscheidungen der Wissensarbeiter:innen stehen, ist es erforderlich, diese Entscheidungen vollständig im Prozessverlauf protokollieren zu können (R23).

AK7: Umfeld

Die Unvorhersehbarkeit von KiPs ist nicht nur innerhalb der Daten- und Prozessentwicklung inherent im Prozessmodell ein Faktor, sondern ist auch gültig hinsichtlich externer Ereignisse. Hieraus resultiert die Anforderung, dass solche externe Ereignisse als ein Auslöser im Prozessmodell definiert werden können, sowohl zur Designzeit (R24) als auch zur Laufzeit (R25).

2.6.4 Forschungslücke

Eine detailliertere Beschreibung der hier aufgeführten Anforderungen kann bei Bedarf nochmals [32] nachgelesen werden, woraus auch hervorgeht, auf welche der vorgestellten Charakteristiken C1-C8 die einzelnen Anforderungen R1-R25 zurückzuführen sind. Im Verlauf dieser Arbeit wurden anschließend zeitgemäße Prozessmodellierungsansätze und Systeme hinsichtlich ihres Erfüllungsgrades der 25 spezifizierten Anforderungen untersucht und beurteilt. Die hierzu ausgewählten Systeme sind eine repräsentative Auswahl [32] der verschiedenen Prozessmanagement-Ansätze, die sich in den letzten Jahren im Rahmen diverser wissenschaftlicher Untersuchungen und Arbeiten herausgebildet haben.

Als Vertreter eines eher aktivitätszentrierten, imperativen Ansatzes mit Konzepten für eine flexible und adaptive Prozessausführung wurden *YAWL* und *ADEPT2* untersucht. Mit *Declare* wurde ein deklarativer Ansatz zur Unterstützung locker strukturierter Prozesse ausgewählt. Als objektbezogener Ansatz wurde *PHILharmonicFlow* und als artefaktzentrierten Ansatz *Artifact* analysiert.

Im Rahmen der durchgeführten Evaluation [32, S. 44-53] wurde beurteilt, inwieweit die untersuchten Ansätze eine Anforderung voll abdecken, diese teilweise oder nicht erfüllen oder ob die Ansätze zur Erfüllung zumindest grundsätzliche Eigenschaften mitbringen. Dabei zeigt sich, dass jeder der Ansätze Stärken und Schwächen in unterschiedlichen Anforderungskategorien hat, dass jedoch keiner der Ansätze die geforderten Anforderungen auch nur zu 50% erfüllen kann. Somit kann keinem dieser Ansätze zugeschrieben

werden, für die Ausführung von KiPs in besonderer Weise geeignet zu sein, woraus sich eine klare Forschungslücke ableiten lässt.

Forschungslücke 1: KiPs

Die etablierten BPM-Ansätze sind (wie in [32] untersucht) nicht oder nur unzureichend für die Ausführung von wissensintensiven Prozessen geeignet.

2.7 Semantische Technologien

Einleitend wurde in Kapitel 2.1 die grundlegende Bedeutung der Begriffe Daten, Information und Wissen eingeführt. Im Anschluss wurde mit den Informationssystemen und daran anknüpfend mit dem Themenbereich der Geschäftsprozesse aufgezeigt, wie vor allem Daten und Informationen in einem spezifischen Anwendungskontext eingebunden sind. In diesem Kapitel wird ein genereller Überblick über semantische Technologien vorgestellt, das Verständnis zu den Begriffen Daten, Information und Wissen vertieft, weitere Begrifflichkeiten eingeführt und damit verbundene Möglichkeiten vorgestellt.

2.7.1 Begrifflichkeiten

Formalisierung aus einer wissenschaftstheoretischen Sicht bedeutet im Grunde „die Generalisierung einer (wissenschaftlichen) Aussage unter Absehung ihrer konkret-empirischen Bezüge“ [7]. So betrachtet ist eine *Formalisierung* auch eine Form der Abstraktion. Anders ausgedrückt: Eine Sache wird formalisiert, indem ihr eine strenge Form gegeben wird oder indem bei ihrer Durchführung eine strenge Form eingehalten wird [124]. Rein pragmatisch betrachtet dient eine Formalisierung dazu, Informationen in eine für Mensch und Maschine verständliche Form zu überführen und eine Mehrdeutigkeit von Begriffen und sprachlichen Äußerungen zu eliminieren.

Damit dient die Formalisierung vornehmlich dazu, ein gemeinsames Verständnis unterschiedlicher Kommunikationsteilnehmer zu gewährleisten, um einen effektiven und korrekten Austausch von Informationen zu ermöglichen

[30, S. 64]. In der Informatik sprechen wir in diesem Fall von einer Konzeptualisierung, deren Ergebnis wir als *Ontologie* bezeichnen. Zumeist wird zur Definition dieses Begriffes Tom Gruber [44] zitiert: „An ontology is an explicit specification of a conceptualization.“ Der Gedanke einer gemeinsamen Nutzung wurde mit der Definition „An ontology is a shared understanding of some domain of interest“ [109] hervorgehoben. Im Folgenden wird mit der Kombination dieser beiden Definitionen nach [104, S. 25] gearbeitet, wonach eine Ontologie eine formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung ist.

Das Ergebnis einer Formalisierung von Wissen wird dann in der Regel eine *Repräsentation* (engl. knowledge representation) genannt, während eine Vorschrift zum Schlussfolgern als *Inferenzierungsmechanismus*, *Inferenzregel* oder kurz *Inferenz* bezeichnet wird [30, S. 23]. Insbesondere die Möglichkeit, mit Hilfe von Schlussfolgerungen auf Basis einer Wissensrepräsentation neues Wissen abzuleiten, stellt einen erkennbaren und expliziten Mehrwert dar und wird mit der Bezeichnung „symbolische KI“ assoziiert.

2.7.2 Arten einer Formalisierung

Damit eine Formalisierung zu einer für alle Beteiligten eindeutigen und interpretierbaren Bedeutung führt, muss ein gemeinsames Modell oder Sprachkonstrukt verabredet werden. Hierzu gibt es verschiedene semantische Modelle, welche sich hinsichtlich ihrer Ausdrucksmächtigkeit unterscheiden, wobei auch von leicht- bis schwergewichtigen Modellen [86, S. 17f.] gesprochen wird. Nachfolgend werden einige der etablierten Modelle aufsteigend hinsichtlich ihrer Mächtigkeit vorgestellt.

- Mit einem *Glossar* wird eine Liste von Begriffen und zugehörigen Erklärungen bezeichnet. Dabei existieren keine Beziehungen zwischen diesen Begriffen.
- Eine *Taxonomie* bezeichnet eine Hierarchie von Begriffen, wobei diese Hierarchie gleichzeitig Beziehungen in Form von Ober- und Unterbegriffen repräsentieren. So betrachtet kennt eine Taxonomie nur einen einzigen Beziehungstyp.
- Ein *Thesaurus* stellt ein Begriffsnetzwerk insbesondere im Bibliothekswesen dar. Ergänzend zur Taxonomie kann hierbei auf weitere fest vordefinierte

nierte Beziehungstypen wie eine Synonymrelation oder eine Ähnlichkeitsrelation zurückgegriffen werden.

- Eine *Topic Map* basiert syntaktisch auf einem XML-Standard und erlaubt die Definition von komplexeren Begriffsdefinitionen mit einem detailreicheren Beziehungsgeflecht.

Schwergewichtige Modelle

Bei den schwergewichtigen Modellen werden die zu definierenden Begriffe oft als Konzepte oder Klassen bezeichnet und je nach Modell werden die aus der objektorientierten Programmierung bekannten Prinzipien der Vererbung unterstützt. Für einen Austausch zwischen verschiedenen Modellen können Namensräume geschaffen und eindeutige Kennzeichner (URIs - Unique Resource Identifier) statt einfacher textueller Bezeichner verwendet werden. Mit Attributen können gezielt Eigenschaften von Konzepten abgebildet werden und selbstdefinierten Beziehungen können gegebenenfalls grundlegende Eigenschaften (wie Symmetrie, Transitivität, ...) zugeordnet werden. Je nach Modell wird auch differenziert zwischen einer konzeptionellen Ebene (Terminologische Ebene, T-Box) und einer Instanzebene (Assertionale Ebene, A-Box) in der einzelne Fakten zum Ausdruck gebracht und mit der konzeptionellen Ebene verknüpft werden können. Nachfolgend werden zwei dieser schwergewichtigen Modelle, RDFS und OWL, vorgestellt.

2.7.3 RDF / RDFS

RDF steht für *Resource Description Framework* und ist ein Standard des World Wide Web Consortiums W3C, der 1999 verabschiedet wurde. Der Standard stellt ein Sprachkonstrukt zur Verfügung, um Metadaten, also beschreibende Informationen, über Internetseiten ausdrücken zu können. Hierdurch können Informationen nicht nur für Menschen, sondern auch für Maschinen verständlich formalisiert werden. Dies folgt der Idee des Semantic Web [13], einer Initiative von Tim Berners-Lee, James Hendler und Ora Lassila, welche das Internet durch die Vernetzung von Webseiten via Hyperlinks nicht nur für den Menschen, sondern mit Hilfe der Metadaten auch für eine Maschine (im speziellen wird von Agenten gesprochen) durchsuchbar macht.

Kern von RDF sind RDF-Konstrukte, welche jeweils eine spezifische Aussage (engl. Statement) über die Welt ausdrücken. Ein solches Statement besteht immer jeweils aus einem Subjekt, einem Prädikat und einem Objekt.

Aufgrund dieser immer konstanten Zusammenstellung von drei Elementen wird auch von einem Tripel gesprochen. So könnte mit einem Tripel beispielsweise zum Ausdruck gebracht werden, dass gewisse Personen in einem Verwandtschaftsverhältnis zueinander stehen oder eine bestimmte Eigenschaft aufweisen. Zur Serialisierung dieser Semantik stehen verschiedene syntaktische Formen zur Verfügung. Als Standardsyntax wird häufig das Format XML/RDF oder XML/OWL verwendet. Eine Alternative mit einer verkürzten und gut lesbaren Form stellt das im nachfolgenden Beispiel verwendete XML/Turtle Format dar.

1. :Anna :ist-Mutter-von :Berta
2. :Berta :ist-Mutter-von :Celine
3. :Celine :geboren-am "10.10.2000"

Anna repräsentiert im Beispiel (1) das Subjekt und über das Prädikat *ist-Mutter-von* wird eine Beziehung zu dem Objekt *Berta* festgelegt. Im Beispiel (2) ist *Berta* das Subjekt und steht in der gleichen Beziehung zu *Celine*. Als Subjekt kann jeweils immer nur eine Ressource verwendet werden, während als Objekt sowohl eine Ressource als auch, wie im Beispiel (3) dargestellt, ein Literal stehen kann. Hierüber kann einer Ressource wie *Celine* gezielt eine Eigenschaft mit einem expliziten Wert zugewiesen werden.

Ein weiteres wesentliches Merkmal von RDF ist, dass eine Ressource (ausgenommen sind Blank Nodes / anonyme Ressourcen) immer eindeutig zu identifizieren sein muss. Dies eröffnet die Möglichkeit einer globalen Vernetzung zwischen Ressourcen, was für das angestrebte Ziel des Semantic Web eine grundlegende Anforderung darstellt. Hierzu wird jede Ressource durch einen innerhalb einer Domain eindeutigen Bezeichner sowie einen eindeutigen Namespace der Domain kombiniert und erhält somit einen *Universal Resource Identifier* (URI).

Das für die Auszeichnung von Webseiten mit Metadaten entworfene Sprachkonstrukt RDF berücksichtigt lediglich existierende Ressourcen und kennt keine Möglichkeit, um grundlegende gemeinsame Konzepte oder Klassen auszudrücken. Hierzu wurde mit RDF-Schema (RDFS) eine Erweiterung zu RDF durch die W3C eingeführt, welche das notwendige Vokabular zur Verfügung stellt, um Ressourcen zu typisieren. Das nachfolgende Beispiel verdeutlicht als eine Erweiterung zu den bereits eingeführten Personen (*Anna*, *Berta*, *Celine*) die Möglichkeiten von RDFS.

1. `:Person rdf:type rdfs:Class`
2. `:Frau rdf:type rdfs:Class`
3. `:Frau rdfs:subClassOf :Person`
4. `:ist-Mutter-von rdf:type rdf:Property`
5. `:ist-Mutter-von rdfs:domain :Frau`
6. `:ist-Mutter-von rdfs:range :Person`
7. `:Anna rdf:type :Frau`

In dem Beispiel wird unter (1) eine Klasse mit dem Bezeichner *Person* und unter (2) eine Klasse mit dem Bezeichner *Frau* definiert. Zeile (3) legt fest, dass *Frau* eine Unterklasse von *Person* ist. In Zeile (4) wird das Prädikat *ist-Mutter-von* definiert und mit (5) und (6) wird festgelegt, dass für dieses Prädikat als Subjekt nur eine Ressource vom Typ *Frau* in Frage kommt, während für das Objekt eine Ressource vom Typ *Person* sein muss. An dieser Stelle sei angemerkt, dass über die *subClassOf* Beziehung jede Instanz vom Typ *Frau* auch automatisch dem Typ *Person* zugehörig ist. Mit Zeile (7) wird dann die Ressource *Anna* der Klasse *Frau* zugewiesen, gleiches kann für *Berta* und *Celine* erfolgen. Hierdurch kann mittels RDFS ein grundlegendes Schema, eine zugrundeliegende Konzeptebene, modelliert werden, während mittels RDF die jeweiligen Instanzen mit Bezug zur Konzeptebene ausgedrückt werden können.

Durch die Erweiterungen mittels RDFS steht das notwendige Vokabular zur Verfügung, um sowohl eine Konzeptionalisierung und damit eine einfache Form einer Ontologie auszudrücken, als auch Instanzen (Ressourcen) mit dieser konzeptionellen Ebene zu verbinden. Dennoch fehlt es an der Ausdrucksmächtigkeit, um komplexere Wissensmodelle darzustellen. So können keine Kombinationen von Klassen (Vereinigung, Schnittmenge, Komplement) ausgedrückt werden, Properties können nicht mit Kardinalitäten belegt werden und Einschränkungen von Properties gelten immer global. Auch fehlt die Möglichkeit, Properties grundlegende Eigenschaften wie Transitivität, Symmetrie oder Invertierbarkeit mitzugeben. Somit fehlt es an grundsätzlichen Sprachkonstrukten, welche für ein leistungsfähiges Schlussfolgern von Sachverhalten notwendig sind.

2.7.4 OWL

Im Hinblick auf diese Schwächen von RDF/RDFS wurde 2004 von der W3C eine standardisierte Ontologiesprache, die Web Ontology Language (OWL) in ihrer ersten Version, spezifiziert. Mit den in den folgenden Jahren gewonnenen Erfahrungen wurde dann 2009 der Standard in einer zweiten Version OWL 2 verabschiedet. Hierbei wurden eine Reihe ontologischer Axiome in Form neuer OWL-Sprachelemente eingeführt, welche eine wesentliche Erweiterung der Ausdrucksmächtigkeit ermöglichen. Zur Differenzierung für unterschiedliche Anwendungsfälle wurden drei Sprachstufen mit einer unterschiedlichen Komplexität und Ausdrucksmächtigkeit eingeführt.

Mit OWL Lite wird nur ein eingeschränkter Umfang an Sprachkonstrukten unterstützt, um ein effizientes Auswerten der modellierten Ontologien zu ermöglichen. OWL DL ist das nächsthöhere Sprachkonstrukt, welches beispielsweise auch boolesche Kombinationen und Kardinalitätsbeschränkungen von Properties erlaubt. Das Akronym DL steht in diesem Fall für Description Logic und kennzeichnet eine entscheidbare Untermenge der Prädikatenlogik, welche auf der Beschreibungslogik SHOIN(D) basiert. Mit OWL Full können alle OWL-Konzepte einschließlich der vollständigen RDFS-Konstrukte verwendet werden, was jedoch zu Konstrukten führen kann, deren Entscheidbarkeit und Vollständigkeit nicht garantiert werden kann.

Anknüpfend an das bereits eingeführte Beispiel wird nachfolgend mit dem erweiterten OWL-Sprachkonstrukt eine zusätzliche ontologische Aussage im Format RDF/XML Syntax getätigt.

```
<owl:ObjectProperty rdf:about="namespace#ist_Großmutter_von">
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  <owl:propertyChainAxiom rdf:parseType="Collection">
    <rdf:Description rdf:about="http://namespace#ist_Mutter_von"/>
    <rdf:Description rdf:about="http://namespace#ist_Mutter_von"/>
  </owl:propertyChainAxiom>
</owl:ObjectProperty>
```

Abbildung 2.6: OWL-Statement in RDF/XML Syntax

Mit dieser Ergänzung wird eine weitere Relation „ist_Großmutter_von“ eingeführt und gleichzeitig wird festgelegt, dass diese gleichgesetzt wird mit einer zweifachen Verkettung der „ist_Mutter_von“ Relation. Diese konzeptionelle Aussage kann nun als Grundlage für einen Inferenzierer genutzt werden, um neues Wissen über die bereits bekannten Fakten zu schlussfolgern. So kann, ohne dass eine explizite Beziehung zwischen *Anna* und *Celine*

besteht, deren verwandschaftliche Beziehung abgeleitet werden. Für männliche Nachkommen müsste für die „ist_Großmutter_von“ Relation noch eine Erweiterung eingeführt werden, auf welche zugunsten einer verkürzten Beschreibung verzichtet wird. Das bisherige Beispiel bietet einen grundsätzlichen wenn gleich oberflächlichen Einblick in die beiden schwergewichtigen Formalisierungsmodelle RDFS und OWL. Einen detaillierten Einblick in den Themenbereich semantische Technologien findet sich beispielsweise in [30].

Sprachregelung

Im weiteren Verlauf wird mit den Begrifflichkeiten *konzeptionelles Wissen*, *terminologisches Wissen* und *T-Box* auf das generalisierte Wissen über Konzepte Bezug genommen. Mit den Begriffen *Instanz* und *Individuum* wird eine konkrete Ausprägung von Konzepten bezeichnet und mit *Faktenwissen* werden mehrere dieser Ausprägungen einschließlich deren Beziehungsgeflecht innerhalb der *A-Box* ausgedrückt. Für die Kombination von *T-Box* und *A-Box* wird dann der Begriff *Wissensbasis* oder ontologisches Wissen verwendet.

2.7.5 SWRL

Gerade mit dem letzten Beispiel aus Abbildung 2.6 wird deutlich, dass innerhalb von OWL nicht nur Statements über Faktenwissen und ontologisches Wissen, sondern auch Regeln zur Identifikation genereller Zusammenhänge (innerhalb der T-Box) im Kontext einer umfassenden Formalisierung von Bedeutung sind. Dabei sind der Ausformulierung von Regeln mittels OWL noch enge Grenzen gesetzt. Deutlich leistungsfähiger ist die von Horrocks et al. vorgeschlagene Semantic Web Rule Language (SWRL) [58]. Auch wenn SWRL zur offiziellen Standardisierung eingereicht wurde, ist es bis heute noch kein offizieller Standard des W3C. Dennoch wurde es in etablierten OWL-Systemen wie Protégé (<http://protege.stanford.edu/>), Pellet (<http://clarkparsia.com/pellet/>) und Hermit (<http://hermit-reasoner.com/>) bereits umgesetzt [30].

Eine SWRL-Regel besteht jeweils immer aus einer Vorbedingung und einer Konsequenz und folgt dem Verständnis, dass falls eine Vorbedingung wahr ist, die formulierte Konsequenz umzusetzen ist. Die mit Abbildung 2.6 eingeführte OWL-Regel kann in SWRL wie folgt ausgedrückt werden.

$$\begin{aligned} \text{ist_Mutter_von}(?F1, ?F2) \quad \wedge \quad \text{ist_Mutter_von}(?F2, ?F3) \\ \rightarrow \quad \text{ist_Gromutter}(?F1, ?F3) \end{aligned}$$

Das Sprachkonstrukt SWRL erlaubt mit dem vorgestellten Fragezeichen die Definition von Variablen innerhalb der Vorbedingung, auf die dann in der Konsequenz zugegriffen werden kann. Darüber hinaus erweitern integrierte Builtins das Sprachkonstrukt um die Möglichkeit zur Auswertung von Literalen, was eine hohe Ausdrucksmächtigkeit zur Berücksichtigung von individuellen Eigenschaften erlaubt. Die vordefinierten Builtins erlauben beispielsweise Vergleichsoperationen, einfache mathematische Berechnungen, boolesche Ausdrücke sowie die Auswertung von zeitlichen Ausdrücken und Textoperationen.

2.7.6 Schlussfolgerungen

Das grundlegende Ziel der Formalisierung und der Definition von Regeln ist, wie bereits eingeführt, die Abbildung von Informationen in einer maschinenlesbaren Form. Nun kann diese formale Repräsentation dazu verwendet werden, um neues Wissen zu den bekannten Fakten zu generieren. Dieser Vorgang wird als Reasoning bezeichnet, wobei es sich um Methoden zum Herleiten neuen Wissens aus einer Wissensbasis handelt [30]. Als Reasoner wird die Komponente oder der Dienst bezeichnet, der diesen Vorgang durchführt.

Vor einem Reasoning muss jedoch noch festgelegt werden, wie der Wissenstand einer Wissensbasis betrachtet werden soll. Wenn man davon ausgeht, dass das Wissen zu einem Sachverhalt vollständig in der Wissensbasis repräsentiert ist, dann spricht man von einer Closed-World-Assumption (CWA). In diesem Fall kann das Reasoning nicht nur auf Grundlage der in der Wissensbasis explizit angegebenen Statements erfolgen, sondern es kann auch davon ausgegangen werden, dass für eine nicht getroffene Aussage das Gegenteil Gültigkeit besitzt. Diese Betrachtungsweise ist jedoch im alltäglichen Gebrauch einer Wissensbasis selten möglich, da in der Regel nicht von einer vollständigen Abbildung eines Sachverhalts ausgegangen werden kann. In diesem Fall wird von einer Open-World-Assumption (OWA) gesprochen, derzufolge die Abwesenheit einer Aussage nicht als eine Negation dergleichen interpretiert werden darf. Grundsätzlich kann festgehalten werden, dass unter der OWA ein weniger weitreichendes Reasoning möglich ist, da von einer Unvollständigkeit der Wissensbasis ausgegangen wird und

demnach nur Schlussfolgerungen getroffen werden können, welche sich vollständig mit dem jeweils bekannten Wissensstand beweisen lassen.

Des Weiteren muss festgehalten werden, dass die verfügbaren Reasoner sich deutlich hinsichtlich ihrer Möglichkeiten zum Schlussfolgern von Zusammenhängen unterscheiden. Dies ist teilweise in der damit verbundenen Laufzeit begründet, da ein vollständiges Schlussfolgern im Rahmen der Möglichkeiten nach OWL-DL und SWRL mitunter NP-schwer sein kann. Ein umfangreicher Vergleich der Leistungsfähigkeit etablierter Reasoner wie Pellet, FACT++, RACER wird in [2] geboten. Die darin beschriebene Fähigkeit einer „Incremental Classification“ ist von besonderer Bedeutung, worauf in Kapitel 6 „Inferenzierung von Prozesszuständen“ nochmals eingegangen wird und weshalb im Rahmen dieser Arbeit der Pellet-Reasoner verwendet wurde.

2.8 Semantische Prozessmodellierung

Das Themengebiet der semantischen Prozessmodellierung befindet sich in der Schnittmenge der beiden eingeführten Themenbereiche. Es vereint die Methoden und Möglichkeiten semantischer Technologien mit dem umfangreichen Forschungs- und Anwendungsgebiet zu Geschäftsprozessen. Im Kern geht es dabei um eine möglichst vollständige Formalisierung der informellen oder semiformalen Prozessmodelle, welche auf Basis der etablierten Prozessmodellierungssprachen (Kapitel 2.4) definiert werden.

In beiden Forschungsbereichen wird die naheliegende Verknüpfung der Themen in der Fachliteratur aufgegriffen. In dem Fachbuch „Business Process Management“ [121] werden die Vorteile einer ontologischen Repräsentation von Prozesselementen insbesondere im Hinblick auf die Möglichkeiten einer automatischen Service-Composition diskutiert. Das Fachbuch „Semantische Technologien“ [30] erläutert an einem Anwendungsfall zur Steuerung von Produktionsprozessen die Vorteile eines semantisch aussagekräftigen Prozessmodells. In allen Fällen wird das grundlegende Ziel verfolgt, eine für Mensch und Maschine eindeutig und verständliche Beschreibung von Prozessmodellen frei von Ambiguitäten (Mehrdeutigkeiten) zu ermöglichen.

Auftretende Ambiguitäten in Prozessmodellen können dabei in zwei grundsätzliche Kategorien aufgeteilt werden. Es kann unterschieden werden in strukturelle und semantische Ambiguitäten [73]. Eine strukturelle Mehrdeutigkeit tritt auf, wenn eine Modellierungssprache keine formale Definition der Modellkonstrukte [37] bietet. Eine semantische Ambiguität tritt auf, wenn eine präzise Definition von Konzepten oder deren Beziehungen untereinander [37, 106] fehlt. So kann es bei einer Begrifflichkeit zu einer unterschiedlichen Interpretation zwischen Prozessteilnehmern kommen oder zu einem bestimmten Konzept werden unterschiedliche Bezeichner verwendet. Thomas et al. [106] sprechen in diesem Fall von einem linguistischen oder begrifflichen Defekt und stellen heraus, dass dies nicht nur im Austausch zwischen menschlichen Prozessteilnehmern ein Problem darstellt, sondern auch eine maschinelle Verarbeitung verhindert.

Während die meisten Methoden zur Prozessmodellierung keine strukturelle Mehrdeutigkeit aufweisen, so ist keiner der etablierten Ansätze frei von semantischer Ambiguität, was Fan et al. in ihrer Arbeit [37] herausstellen.

2.8.1 SBPM-Lifecycle

Im Rahmen diverser Publikationen werden unterschiedlichste Anwendungs- und Nutzungsszenarien im Zusammenhang mit semantischen Prozessmodellen aufgegriffen und diskutiert. Diese adressieren jeweils einzelne Aspekte innerhalb des in Abbildung 2.2 vorgestellten BPM-Lebenszyklus. An die Vorarbeiten anknüpfend führt Wetzstein et al. in [123] ein auf die semantische Prozessmodellierung abgestimmtes Modell eines Lebenszyklus ein. Hierbei wird die Bezeichnung SBPM, eingeführt in [53], aufgegriffen und zu einem SBPM-Lifecycle erweitert.

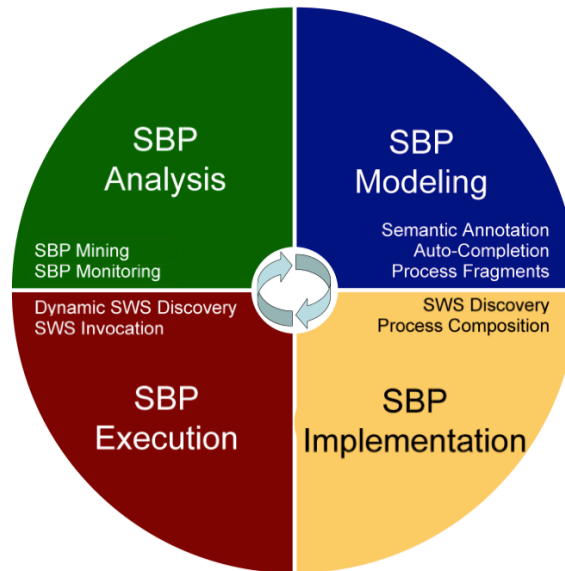


Abbildung 2.7: SBPM-Lebenszyklus [123]

Grundsätzlich werden bei dem in Abbildung 2.7 dargestellten SBPM-Lebenszyklus nach [123] die gleichen vier Phasen wie bei [121, S. 11] identifiziert. Dabei wird in [123] auf die jeweiligen Besonderheiten, Anforderungen und Vorteilen eines semantischen Geschäftsprozessmodells eingegangen, welche im Folgenden kurz beschrieben werden.

SBP-Modeling

In der ersten Phase wird ein vollständiges und eindeutig semantisch annotiertes Prozessmodell vorausgesetzt. Der angestrebte Nutzen liegt zum einen in der Möglichkeit, für durchzuführende Aktivitäten aufgrund der semantischen Beschreibung autonom geeignete Semantic Web Service (SWS) identifizieren zu können. Des Weiteren wird eine einfachere und bessere Wiederverwendbarkeit von Prozessfragmenten aufgrund der vollständigen Formalisierung avisiert. Mit der Möglichkeit einer Auto-Completion soll zudem der Prozessdesigner bei seiner Arbeit unterstützt und der zeitintensive Vorgang der Prozessmodellierung vereinfacht werden.

SBP-Implementation

In der zweiten Phase wird die Umsetzung eines Prozessmodells hin zu einer ausführbaren Prozessdefinition adressiert. Hierbei kann die semantische Repräsentation eines Prozessmodells für KI-gestützte Prozess-Planungsabläufe und für eine automatische Service-Komposition ausgenutzt werden. Zudem unterstützt ein SBP-Modell auch die SBP-Implementation durch eine automatisierte Auswahl von geeigneten Semantic Web Services.

SBP-Execution

In der Phase der SBP-Ausführung werden drei Ebenen adressiert und die Vorteile einer auf SBPM basierenden Prozessinstanz beschrieben. Auf der Ebene der „Implementation“ findet die eigentliche Instanziierung einer Prozessinstanz auf Grundlage einer Prozessdefinition statt. Die semantische Beschreibung kann hier zur Identifikation geeigneter Ressourcen wie SWS dienen. Auf der Ebene „Registry“ wird die jeweilige Prozessinstanz auf einer Management-Ebene eingebucht und durch die semantische Kennzeichnung präzise gegenüber externen Stakeholdern beschrieben. Auf der Ebene „Consumer“ findet der Austausch mit den Anwender:innen und die Aktivierung und die Interaktion mit SWS statt.

SBP-Analysis

In der vierten Phase geht es erwartungsgemäß um das Lernen von durchgeführten Prozessinstanzen und das Verbessern bestehender Prozessmodelle oder Prozesssegmente. Damit verbunden sind eine Reihe an Funktionalitäten, welche von der formalen Beschreibung von Prozessmodell, Prozessdefinition und Prozessinstanz profitieren können. Im Einzelnen wird dabei, auch mit Blick auf eine prozessdesignende Person, auf Funktionalitäten geblickt, welche Unterstützung durch eine semantische Prozesssuche, eine semantische Konformitätsprüfung, semantische Organisationsanalyse, semantische Performanceanalyse oder ein semantisches Auditing bieten.

Anmerkung

In der Beschreibung des SBPM-Lebenszyklus nach [123] wird sich im Wesentlichen darauf beschränkt, welche Vorteile eine semantische Annotation zur Identifikation und zur Ausführung vorwiegend externer Ressourcen wie SWS und Anwender:innen haben kann. Es wird jedoch nicht identifiziert, welche unmittelbare Rolle ein formalisiertes Prozessmodell im Hinblick auf

eine Prozessausführung haben kann, sei es über konzeptionelles Wissen wie domainenspezifisches Expertenwissen oder über Faktenwissen im Kontext der auszuführenden Prozessinstanz. Zudem sind alle Erläuterungen und Beschreibungen einem kontrollflussorientierten Modellierungsansatz zuzuordnen, wodurch nur eine Teilmenge der Ansätze zur Prozessmodellierung berücksichtigt sind.

2.8.2 Auszug relevanter Arbeiten

Im Rahmen des eingeführten SBPM-Lifecycle können mit Bezug zu den einzelnen Phasen weitere relevante Arbeiten identifiziert werden, wozu auszugswise nachfolgend einige vorgestellt werden.

Die Arbeit von Betz et al. [16] widmet sich der Unterstützung der Prozessdesigner:innen in der ersten Phase des Lebenszyklus. Im Zentrum steht dabei die semantische Beschreibung eines Prozessmodells mittels OWL/DL mit dem Ziel, die Mehrdeutigkeit innerhalb von Petri-Netze zu beseitigen. Zur Unterstützung im Designprozess werden Verfahren entwickelt, um eine Aussage zu Ähnlichkeiten zwischen SBP-Modellen auf Basis syntaktischer, linguistischer oder struktureller Merkmale treffen zu können.

Heinrich et al. [52] stellt mit SEMPA (SEMantic based Planing Approach) einen Ansatz vor, um Prozessmodelle auf Basis einer semantisch beschriebenen Prozessbibliothek mit Aktivitäten automatisch planen zu lassen. Hierzu werden zunächst grundlegende Anforderungen an einen der Aufgabenstellung entsprechenden Planungsprozess beschrieben und ähnliche Arbeiten in diesem Umfeld werden hinsichtlich ihrer Erfüllung dieser Anforderungen analysiert. Die semantisch beschriebene Prozessbibliothek wird dann von einem maßgeschneiderten SEMPA-Planer zur Planung eines Kontrollflusses verwendet.

Mit Poba (Process Ontology Based Approach) stellen Fan et al. [37] ebenfalls einen Ansatz vor, der die Prozess-Planungsphase adressiert. Dabei wird zunächst eine Methodologie eingeführt, um die Designer:innen mit einer Referenz oder Best-Practice Anleitung zu unterstützen. Zudem wird das Knowledge-Management von Unternehmen und Branchen fokussiert, um deren Verbindung in die Prozessabläufe bereits im Planungsverlauf besser berücksichtigen zu können. Grundsätzlich widmet sich Poba der manuellen und perspektivisch teilautomatisierten Planung von BPMN-

Prozessmodellen und führt eine empirische Evaluation durch, deren Nutzen auch auf die Disambiguität durch SBPM zurückgeführt werden kann.

Einen ähnlichen Weg verfolgen auch Thomas und Fellmann [106], bei dem ausgehend von BPMN und EPC jedoch eine sprachneutrale Methodik angestrebt wird. Dies gelingt nur begrenzt im Rahmen des kontrollflussorientierten Prinzips, unterstreicht jedoch ebenfalls die Vorteile einer formalisierten Beschreibung von Prozesselementen und Prozessstrukturen.

2.8.3 Forschungslücken

Der Wert einer tiefgreifenden Formalisierung von Prozessmodellen im Hinblick auf eine Disambiguität wurde von vielen Forschungsarbeiten aufgegriffen und belegt. Auch die enge Beziehung zwischen datenorientierten Prozessen, deren Anforderungen und einer Verbindung zu einer semantischen und einheitlichen Repräsentation dieser Daten auf Basis von Ontologien und semantischen Technologien wird in einer Reihe von Arbeiten [74, 30, 116, 121, 98] benannt. Die identifizierten Arbeiten [106, 37, 52, 16] fokussieren jedoch jeweils kontrollflussorientierte BP-Modelle und greifen nicht die in Kapitel 2.3.2 vorgestellten datenorientierten Ansätze auf. Abseits der eigenen Vorarbeiten des Autors [94, 93, 68] ergibt sich hier als Forschungslücke:

Forschungslücke 2: SBPM + datenorientierte Ansätze

Die Vorteile eines datenorientierten Prinzips zur Prozessmodellierung und die Möglichkeiten einer semantischen Prozessmodellierung wurden bislang nicht in einem gemeinsamen Ansatz kombiniert und deren erwarteter Nutzen in einem realisierten System evaluiert.

Ein weiterer wesentlicher Baustein einer semantischen Wissensrepräsentation liegt in der Möglichkeit, mittels Inferenzierung neues Wissen zu bekannten Fakten zu generieren. Die Forschungsarbeiten greifen diese Möglichkeiten teilweise auf [106, 37, 52], verwenden diese jedoch nur im Bezug auf die manuelle oder automatisierte Planung von Prozessmodellen. Ein unmittelbarer Einfluss von geschlussfolgertem Wissen in einer laufenden Prozessinstanz wird in den bekannten Arbeiten nicht adressiert. Abseits der eigenen Vorarbeiten des Autors [94, 93, 68] ergibt sich hier als Forschungslücke:

Forschungslücke 3: Inferenzierung von Prozesszuständen

Die Möglichkeiten, mittels Inferenzierung einen unmittelbaren Prozesseinfluss in Form von geschlussfolgerten Prozesszuständen auszuüben, ist bislang mit keinem Ansatz adressiert und über ein realisiertes System evaluiert worden.

2.9 Prozessvisualisierung

Die Modellierung umfangreicher Geschäftsprozesse kann zu sehr großen und unübersichtlichen Prozessmodellen [63] führen. Ein erster Ansatz zur Beherrschung solch komplexer Modelle liegt in einer graphischen Darstellung, frei nach dem Sprichwort: „Ein Bild sagt mehr als tausend Worte“. Grundsätzlich helfen Visualisierungen dem Nutzer, das Unsichtbare zu sehen, denn visuelle Darstellungen machen Zusammenhänge sichtbar und ermöglichen auf diesem Weg ein tieferes Verständnis eines Sachverhalts [122]. Geschäftsprozesse werden dabei zumeist mit Hilfe von spezialisierten Notationen in Form einer graphischen Repräsentation beschrieben [3], welche die zugrundeliegenden Prozesselemente und die Prozesslogik reflektieren. Der am weitesten etablierte Standard BPMN [97] spezifiziert neben den Modellierungsregeln eine ebensolche Notation.

Das grundsätzliche Ziel dieser graphischen Repräsentationen ist es, die mitwirkenden Personen dabei zu unterstützen, ein Geschäftsprozessmodell besser zu verstehen. Dabei wird in unterschiedliche Personengruppen mit unterschiedlichen Anforderungen unterschieden. *Prozessdesigner:innen* widmen sich der Modellierung von Geschäftsprozessen und benötigen zumeist ein sehr detailliertes und vollständiges Abbild eines Modells. *Prozessanwender:innen* tragen bei der Ausführung einer Prozessinstanz durch ihre eigene Mitwirkung bei und profitieren von einer auf die eigene Aufgabenstellung ausgerichteten Darstellung. *Prozessmanager:innen* möchten vor allem einen Überblick über eine Menge von Prozessinstanzen erhalten und benötigen hierfür eine abstraktere Darstellung von Prozesszuständen und Prozessfortschritten.

Geschäftsprozesse werden in der Regel als gerichtete Graphen visualisiert, um den Fluss von Ressourcen, Aufgaben und Zeit sichtbar zu machen [66]. Im Allgemeinen gibt es verschiedene Graphen-Layout-Algorithmen [105, 56] sowie auf Geschäftsprozessgraphen zugeschnittene Layout-Ansätze [34, 4]. Dabei adressieren diverse Studien die grundlegende Herausforderung, innerhalb einer Graphdarstellung die Informationen eines komplexen Prozesses abzubilden. So werden beispielsweise die Möglichkeiten untersucht, mithilfe einer 3D-Visualisierung die umfangreichen Details von Geschäftsprozessen [99, 15, 24] sichtbar zu machen. Vor- und Nachteile einer 2D- oder 3D-Visualisierung werden dabei intensiv diskutiert. Einerseits bietet die dritte Dimension die Möglichkeit, zusätzliche Informationen zu transportieren [24], andererseits zeigen Studien, dass bereits in einer 2D-Darstellung die

Menge an Information, die auf verständliche Weise integriert werden kann, begrenzt ist [15].

Animationen sind ein weiterer Weg um zu visualisieren, wie Geschäftsprozesse ausgeführt werden [95] und dieser Ansatz wird in Tools wie ARIS oder IBM WBI Modeler bereits eingesetzt. Auch dieser Aspekt wird kontrovers diskutiert. Einzelne Forschungsstudien zeigen auf, dass Animationen die Nutzer:innen dabei unterstützen können, Änderungen nachzuvollziehen und den Prozessablauf leichter zu verstehen [25, 35]. Im Gegensatz dazu weisen Beispiele aber auch darauf hin, dass mit der Animation eine kognitive Belastung einhergeht, was sich nachteilig auf das Verständnis der Anwender:innen auswirkt [10].

Die Komplexität von Geschäftsprozessen beruht letztlich auf der Vielzahl an Informationen wie Kontrollflüsse, Datenflüsse und Daten zu Prozesselementen und zusätzlichen im Prozess eingebundenen Ressourcen. Zur Darstellung entsprechender Prozessgraphen eröffnen sich eine Vielzahl unterschiedlicher Ansätze, mit Hilfe derer die Visualisierung komplexer Zusammenhänge umgesetzt und die Nachvollziehbarkeit für die Anwender:innen verbessert werden kann. Bobrik et al. [20] differenzieren hierzu die Freiheitsgrade zur Gestaltung einer Prozessvisualisierung (PV) gemäß der in Abbildung 2.8 dargestellten Struktur.

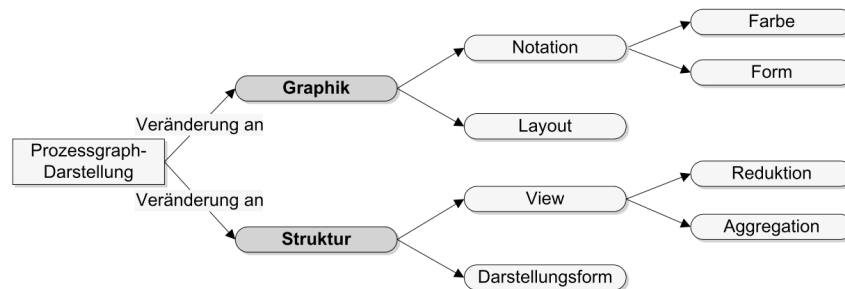


Abbildung 2.8: Freiheitsgrade der Gestaltung einer Prozessvisualisierung [20]

Demnach wird zunächst unterschieden in die graphischen und die strukturellen Aspekte einer Prozessgraph-Darstellung. Zu den graphischen Aspekten gehört die Notation, also die zur Darstellung von Prozesselementen und Zuständen verwendeten Symbole. Durch Veränderungen an Formen und Farben erzielt man eine unterschiedliche Wahrnehmung bei den Nutzer:innen. Über das Layout wird die Positionierung verschiedener Prozesselemente festgelegt. Das daraus resultierende graphische Erscheinungsbild prägt sich bei den Betrachter:innen ein und wird auch als Mental Map [85] bezeichnet. Diesem Aspekt wird eine zentrale Bedeutung hinsichtlich der Wiedererkennung von und Orientierung in Prozessen beigemessen.

Als einen strukturellen Aspekt wird die View betrachtet, womit festgelegt wird, welche Elemente im Rahmen einer Darstellung in welcher Art und Weise dargestellt werden. So kann die Komplexität einer Prozessdarstellung reduziert werden, indem Veränderungen an der View vorgenommen werden. Hierzu können Elemente aus einer Darstellung entfernt oder mehrere Elemente aggregiert werden. Darüber hinaus kann die Darstellungsform variieren, sodass Prozessinformationen auch in Form von Tabellen oder Projektplänen präsentiert werden.

Im Rahmen dieser Arbeit werden vor allem strukturelle Darstellungsveränderungen fokussiert, weshalb nachfolgend die damit in Verbindung stehenden Vorarbeiten besonders betrachtet werden.

Im Rahmen des *Proviado* Projektes wurden umfangreiche und komplexe Prozesse aus der Automobilbranche aufgegriffen und die Nachvollziehbarkeit der kontrollflussorientierten Abläufe für die Anwender:innen wurde als eine zentrale Herausforderung identifiziert [95]. Im Besonderen wurde auf Prozessdefinitionen, Prozessinstanzen und die dynamischen Veränderungen während einer Prozessausführung eingegangen. Als Lösungsansatz wurden zunächst vor allem Strategien für das Layouting erarbeitet.

In den daran anknüpfenden Arbeiten [21, 20] wurden SESE-Blocks (Single-Entry-Single-Exit) als eine Möglichkeit identifiziert, strukturelle Veränderungen innerhalb der Prozessgraphdarstellung vorzunehmen. Hierzu wurden die Reduktion und die Aggregation als zwei mögliche Operationen vorgestellt, um solche SESE-Blocks zu verändern. Die Motivation entspringt dabei der Tatsache, dass Anwender:innen mit unterschiedlichen Rollen in einem Prozess mitwirken und sich daraus der Bedarf an einer auf die Aufgabenstellung ausgerichteten Prozessdarstellung ableitet.

Schumm et al. verfolgen mit ihrem Ansatz einer State Propagation [100] einen anderen Ansatz zur Vereinfachung komplexer Prozessvisualisierungen.

Hierbei wird eine abstrakte Prozessdarstellung (High-Level Notation) verwendet und mit den Zuständen eines wesentlich detaillierter ausgearbeiteten Prozessmodells (Low-Level Notation) verbunden. Auf diesem Wege wird der Prozessfortschritt einer Prozessinstanz durch eine State-Propagation in die abstraktere und damit einfachere Prozessdarstellung zurückgespiegelt.

In weiteren Arbeiten zu Proviado wurde ein Ansatz entwickelt, wie strukturelle Vereinfachungen wie Reduktion und Aggregation auf der Ebene einer Prozessdefinition vorbereitet und anschließend auf Prozessinstanzen übertragen werden können. Hierbei hat man sich damit befasst, dass sich eine Prozessinstanz in unterschiedlichen Prozesszuständen befinden kann und eine Vereinfachung der Prozessdarstellung den jeweiligen Zustand eines einzelnen Prozesselementes berücksichtigen sollte.

Mit dem Projekt proView wurde an den Vorarbeiten von Proviado angeknüpft und die Veränderungen einer Prozessdarstellung [63] noch stärker auf die Anwender:innen ausgerichtet. Im Besonderen wird hierbei die Parametrisierbarkeit einer anwenderorientierten Prozessvisualisierung für eine kontrollflussorientierte Darstellung unterstrichen. Darüber hinaus wird das Verfahren auch für eine datenflussorientierte Darstellung [64] adaptiert, wenn gleich der Datenfluss dem zugrundeliegenden Kontrollfluss folgt und lediglich eine alternative Darstellungsform bietet.

Die grundlegende Bedeutung einer auf die Anwender:innen ausgerichteten Adaption einer Prozessvisualisierung (aPV) wird durch die genannten und weiteren [20, 62] Vorarbeiten nochmals deutlich. Auch die Notwendigkeit, den Prozessstatus bei einer aPV zu berücksichtigen wird durch diese Vorarbeiten unterstrichen. Allerdings waren diese Studien jeweils auf die etablierten, kontrollflussorientierten Ansätze wie BPMN oder EPC ausgerichtet, woraus sich die vierte Forschungslücke ergibt.

Forschungslücke 4: aPV

Die Möglichkeiten und Vorteile einer Adaption und damit vor allem einer Vereinfachung komplexer Prozessmodelle wurde bislang nicht mit einem deklarativen, datenorientierten Ansatz kombiniert und über ein realisiertes System evaluiert.

Mit Kapitel 7 wird die Forschungslücke gezielt aufgegriffen, um durch verschiedene Methoden der Adaption eine Vereinfachung einer komplexen Prozessdarstellung eines deklarativen, datenorientierten Prozessmodells zu erzielen. Dies ist von besonderer Bedeutung im Hinblick auf die angestrebte Flexibilität. Wie innerhalb der eingeführten Charakteristiken von Geschäftsprozessen 2.3.2 dargelegt wurde, sind Flexibilität und Prozessunterstützung zunächst als gegenläufige Eigenschaften [84] zu betrachten. Mit Hilfe einer aPV soll diesem Effekt entgegengewirkt werden, indem die Adaption auch gezielt als ein Werkzeug zur Prozessunterstützung verwendet wird.

3 Problemstellung und Lösungsansatz

Im Zentrum dieser Arbeit steht, wie bereits in der Einleitung eingeführt, die Ausführung wissensintensiver Prozesse und die Frage, wie die Ausführung dieser besonderen Form der Geschäftsprozesse durch einen geeigneten BPM-Ansatz unterstützt werden kann. Aufbauend auf den in Kapitel 2.6 eingeführten Vorarbeiten werden nachfolgend drei grundlegende Kernanforderungen an KiPs definiert. Anschließend wird ein Lösungsansatz konzipiert, der auf den in Kapitel 2.3 eingeführten Grundprinzipien aufbaut und im Zusammenhang mit den identifizierten Forschungslücken betrachtet wird. Die Realisierung des Lösungsansatzes stützt sich dann auf die Umsetzung von drei zentralen Artefakten, deren Einfluss auf die definierten Kernanforderungen nochmals zurückgespiegelt und beschrieben werden.

3.1 Kernanforderungen von KiPs

Innerhalb der Grundlagen wurden in Kapitel 2.6 die generellen Charakteristiken zu wissensintensiven Prozessen eingeführt und erläutert. Demnach sind KiPs *wissensgetrieben, kollaborationsorientiert, unvorhersehbar, sich entwickelnd, zielgetrieben, ereignisgetrieben, regelgetrieben* und *nicht wiederholbar*. Di Ciccio et al. [32] überführt diese Charakteristiken in 25 sehr detaillierte Anforderungen, die sich zusammenfassen und in *drei Kernanforderungen* für KiPs überführen lassen. Diese finden sich mit ähnlichen Bedeutungen und ähnlichen Bezeichnungen in weiteren Arbeiten [43, 50, 110, 111] zu KiPs wieder. Zur Harmonisierung wird der gemeinsame Nenner dieser Arbeiten aufgegriffen und die Kernanforderungen werden mit einer einheitlichen Begrifflichkeit im Sinne der nachfolgenden Definition verwendet.

1. *Flexibilität* wird entsprechend der Definition in Kapitel 2.5.1 fortan als die Fähigkeit verstanden, mit vorhersehbaren sowie unvorhersehbaren Veränderungen innerhalb eines Geschäftsprozesses umzugehen.

2. *Adaptivität* wird entsprechend der erweiterten Definition in Kapitel 2.5.2 fortan als die Fähigkeit verstanden, einen Prozess bzw. dessen Prozessdarstellung an sich entwickelnde und verändernde Prozesszustände sowie an die individuellen Anforderungen mitwirkender Personen anzupassen.
3. *Zielorientierung* wird entsprechend der Definition in Kapitel 2.5.3 fortan als die Fähigkeit verstanden, Veränderungen von Prozesszielen während der Prozessausführung zu berücksichtigen und die Anwender:innen bei der Erreichung dieser Ziele zu unterstützen.

Diese Kernanforderungen reflektieren teilweise die in der Literatur [111, 32, 43, 110] als wesentlich deklarierten Anforderungen an einen für KiPs geeigneten BPM-Ansatz. Andere Anforderungen wie beispielsweise die Unterstützung von Anwender:innen bezüglich einer kollaborativen Arbeitsweise sind zwar grundsätzlich relevant, gehören aber nicht zum Fokus dieser Arbeit. Entsprechend spiegeln die gewählten und definierten drei Kernanforderungen die Ausrichtung dieser Forschungsarbeit wider.

Innerhalb des Kapitels 2.6 wurde auch aufgezeigt, dass keiner der etablierten BPM-Ansätze sich umfassend für die Ausführung von KiPs eignet [32], was als Forschungslücke 1 spezifiziert wurde. Diese Arbeit, deren Ausrichtung durch die Forschungsfragen innerhalb der Einleitung bereits grob definiert wurde, positioniert sich innerhalb dieser Forschungslücke.

3.2 Lösungsansatz

In Kapitel 2.3.2 wurden die wesentlichen Grundprinzipien der Prozessmodellierung eingeführt, wovon einige mit Blick auf die Kernanforderungen vorteilhafte Eigenschaften aufzeigen.

- Im direkten Vergleich zwischen einem *imperativen* und einem *deklarativen* Prozessmodellierungsansatz wird deutlich, dass sich der *deklarative Ansatz* zur Erfüllung der Kernanforderung hinsichtlich Flexibilität eignet.
- Im direkten Vergleich zwischen einem kontrollflussorientierten und einem datenorientierten Ansatz wird deutlich, dass der *datenorientierte Ansatz* ebenfalls eine flexible Prozessausführung gestattet. Gleichzeitig wird die Ausrichtung des Prozesses an einen durch die Daten reflektierten Prozesszustand ermöglicht und damit die Neuausrichtung auf sich verändernde Prozessziele (Zielorientierung) begünstigt.

- Eine *semantische Prozessmodellierung* statt einer semi-formalen Umsetzung bildet die Grundlage, um neue Fakten innerhalb einer Prozessinstanz zu schlussfolgern. In der späteren Ausarbeitung in Kapitel 6 wird gezeigt, wie dies für eine flexible Prozessausführung genutzt werden kann, indem zu jedem Zeitpunkt der Prozesszustand an den jeweils aktuellen Prozessdaten ausgerichtet werden kann. Dabei wird auch gezeigt werden, wie sich verändernde Prozessziele identifizieren lassen und in Folge dessen eine Zielorientierung innerhalb der Prozessausführung erreicht werden kann.

Der Lösungsansatz liegt in der Kombination der drei Grundprinzipien mit dem Ziel, die jeweiligen vorteilhaften Eigenschaften in einem neuen Ansatz zu vereinen. Da die semantische Prozessmodellierung auf einer Konzeptionalisierung (einer Ontologie) aufsetzt und in diesem Rahmen innerhalb eines zu definierenden Metamodells ein deklarativer und datenorientierter Modellierungsansatz angestrebt wird, wird der neue Ansatz nachfolgend als (engl.:ontology- and data-driven business process) (ODD-BP) bezeichnet. Zudem kann festgestellt werden, dass sich, mit Blick auf die in Kapitel 2.8 untersuchten Vorarbeiten und den identifizierten Defiziten, der neue ODD-BP Ansatz innerhalb der beschriebenen Forschungslücken 1, 2 und 3 positioniert.

Die Fähigkeit der Adaptivität im Sinne der etablierten Definition (Kapitel: 2.5.2), also der Anpassung des Prozessmodells um Eigenschaften wie Flexibilität zu erwirken, ist bei einem deklarativen Modellierungsansatz nicht erforderlich. Anders verhält es sich bei Adaptivität im Sinne der eingeführten erweiterten Definition. Demnach wird Adaptivität als die Fähigkeit verstanden, einen Prozess bzw. dessen Prozessdarstellung an sich entwickelnde und verändernde Prozesszustände sowie an die individuellen Anforderungen mitwirkender Personen anzupassen. In Kapitel 2.9 wurden Ansätze vorgestellt, welche im Rahmen dieser Arbeit als Vertreter einer adaptiven Prozessvisualisierung (aPV) bezeichnet werden. Dieses Prinzip von aPV wird als eine Erweiterung der im ODD-BP Ansatz bereits kombinierten Grundprinzipien hinzugefügt. Da in bisherigen Studien zur Adaption einer Prozessdarstellung jeweils nur kontrollflussorientierte Prinzipien untersucht wurden, ist diese Erweiterung innerhalb des als Forschungslücke 4 beschriebenen Defizits positioniert.

3.3 Artefakte

Die Kombination der Ansätze zu einem neuen, als ODD-BP bezeichneten Ansatz repräsentiert die dieser Arbeit zugrundeliegende Strategie. Nachfolgend wird auf die Artefakte eingegangen, welche, dem designorientierten Forschungsansatz folgend, zur Umsetzung der Strategie und damit zur Erfüllung der Aufgabenstellung konzipiert werden.

3.3.1 A1: Metamodell

ODD-BP steht, wie eingeführt, für ein *deklaratives, datenorientiertes und semantisch beschriebenes* Prozessmodell. Hierbei sollen alle wissenstragenden Segmente in eine einheitliche Wissensbasis integriert werden, was auch das grundsätzliche Sprachkonstrukt einschließt, mittels dessen Prozesse für eine bestimmte Aufgabenstellung modelliert werden können. Dieses Sprachkonstrukt kann als Metamodell bezeichnet werden, was definiert ist als „die Rahmen, Regeln, Einschränkungen, Modelle und Theorien, die für die Modellierung einer vordefinierten Klasse von Problemen anwendbar und nützlich sind“ [40, 77]. Darin werden alle Prozesselemente und ihre möglichen Beziehungen, die diese miteinander eingehen können, definiert. Dadurch wird letztlich auch der grundsätzliche Charakter eines prozessführenden Systems festgelegt, indem es durch die möglichen Beziehungen zwischen den Prozesselementen beispielsweise einen eher strikten kontrollflussorientierten Charakter, einen ereignisgetriebenen Charakter oder eben einen deklarativen, datenzentrierten Charakter ausprägt. Das erste Artefakt dieser Arbeit ist somit das *Metamodell*, das naheliegenderweise in die Kategorie der Modelle einsortiert werden kann und in Kapitel 5 vorgestellt wird.

3.3.2 A2: Inferenzieren von Prozesszuständen

Die Umsetzung der Kernanforderungen wie Flexibilität erfordern eine geeignete Methodik zur Steuerung einer Prozessausführung. Über das Metamodell wird nach Umsetzung von A1 bereits ein deklarativer und datenorientierter Charakter des Prozessmodells vorgegeben. Hierfür ist die etablierte Methodik, wie sie bei einem kontrollfluss-orientierten Verfahren verwendet wird, ein sich sequentiell entwickelnder Prozesszustand, ungeeignet. Stattdessen wird ein Verfahren benötigt, das einen beliebigen und auch ungelplanten Prozessbeitrag unmittelbar berücksichtigen und in einen neuen Prozesszustand überführen kann. Der Zustand eines Prozesses kann dabei als die Summe der individuellen Zustände einzelner Prozesselemente betrachtet

werden, was beispielsweise die Ausführbarkeit von Aktivitäten oder auch die Relevanz von Prozesselementen beinhaltet. Im Rahmen dieser Arbeit wird deshalb das Ziel verfolgt, den individuellen Zustand einzelner Prozesselemente zu jedem Zeitpunkt schlussfolgern zu können. Eine sequentielle Prozessentwicklung wird hierbei durch einen alternativen Ansatz ersetzt, den man als eine permanente Prozessplanungsprozedur bezeichnen könnte. Das entsprechende Artefakt ist dann eine Menge an Regeln, welche das *Inferenzieren der Zustände* der Prozesselemente gestattet, ein Artefakt aus der Kategorie Methodik. Diesem Artefakt wird sich in Kapitel 6 gewidmet.

3.3.3 A3: Adaptive Prozessvisualisierung

Die identifizierten Anforderungen adressieren bislang lediglich die inneren Mechanismen eines für KiPs geeigneten Ansatzes, unberücksichtigt sind jedoch bislang die Anwender:innen. Ein Ansatz der eine weitreichende Flexibilität und Zielorientierung aufweist, führt auch zu angepassten Bedürfnissen auf Seiten der mitwirkenden Personen, da diese in die Lage versetzt werden müssen, von den neuen Möglichkeiten Gebrauch zu machen. Während bei kontrollflussorientierten Verfahren vor allem einzelne oder wenige ausführbare Aktivitäten für die Anwender:innen in Frage kommen, kann die angestrebte Flexibilität nur durch vielfältige Möglichkeiten zur Prozesseinwirkung erreicht werden. Eine umfassende Darstellung aller Möglichkeiten erscheint jedoch auch ungeeignet, da hierbei nicht zwischen wichtigen und weniger wichtigen Prozessbeiträgen unterschieden werden kann, wodurch eine Unterstützung entsprechend gering ausfallen würde. Sinnvoll erscheint hingegen eine Prozessvisualisierung, welche sich adaptiv verhält und die Ausführbarkeit und Relevanz von Prozesselementen berücksichtigt, um den Anwender:innen eine substanzielle Prozessunterstützung bieten zu können. Das Inferenzieren einer generellen Relevanz von Prozesselementen (z.B. der Ausführbarkeit von Aktivitäten) ist hierzu bereits ein vielversprechender Anknüpfungspunkt, allerdings repräsentiert dies nur eine Relevanz von allgemeiner, für alle Personen gleicher Natur. Jede(r) Nutzende hat jedoch hinsichtlich der eigenen Rolle auch eine individuelle Einschätzung von Relevanz, welche sich in einer entsprechenden individualisierten, adaptiven Prozessvisualisierung widerspiegeln sollte. Das dritte Artefakt ist somit eine adaptive Prozessvisualisierung (aPV), welche ebenfalls der Kategorie Methodik zugeordnet werden kann und in Kapitel 7 eingeführt wird.

3.3.4 A4: ODD-BP System

Die vorgestellten Artefakte A1 - A3 widmen sich konzeptionell der in FF1 adressierten Fragestellung und adressieren in unterschiedlichem Maße die eingeführten Kernanforderungen. Hierauf wird nachfolgend nochmals im Detail eingegangen. Für die im Rahmen dieser Arbeit erforderliche Evaluation wird neben den entwickelten Artefakten aus dem Bereich der Modelle und Methodik noch ein weiteres Artefakt - A4 - erforderlich. A4 widmet sich dabei der prototypischen Umsetzung der in A1, A2 und A3 geschaffenen Artefakte in Form eines evaluierbaren Systems. Hierzu wird in Kapitel 8 die Realisierung eines Systems auf Basis einer Client-Server Architektur vorgestellt. Dies umfasst im ersten Schritt zunächst eine generische Systemumsetzung ohne eine domänenspezifische Ausrichtung. Erst im Anschluss erfolgt ab Kapitel 8.2 eine Anpassung auf den im Rahmen dieser Arbeit adressierten Anwendungsbereich. Im Zuge dieser Ausrichtung auf die Domäne Leitstelle, konkret der Notrufbehandlung durch Disponent:innen, wird zum einen ein Rahmen geschaffen, der einen Vergleich zwischen dem Status-Quo mit dem neuen Ansatz nach ODD-BP erlaubt. Darüber hinaus wird das System mit Expertenwissen angereichert, welches bei der Ausführung eines KiPs einen domänenspezifischen Beitrag leisten kann. Das vierte Artefakt kann somit innerhalb des designorientierten Forschungsansatzes in die Kategorie der Instanziierung eingeordnet werden.

3.4 Bezug zu den Forschungsfragen

Entsprechend der für KiPs definierten Kernanforderungen werden, wie dargestellt, drei Grundprinzipien der Geschäftsprozessmodellierung mit vorteilhaften Eigenschaften in einem kombinierten Ansatz vereint und mit der Bezeichnung ODD-BP geführt. Dies zielt unmittelbar auf die Beantwortung der in Kapitel 1 eingeführten ersten Forschungsfrage (FF1) ab: *Kann aus den drei dargelegten Grundprinzipien ein einheitlicher Ansatz entwickelt werden, der die Kernanforderungen wissensintensiver Prozesse hinsichtlich Flexibilität, Adaptivität und Zielorientierung erfüllt?*

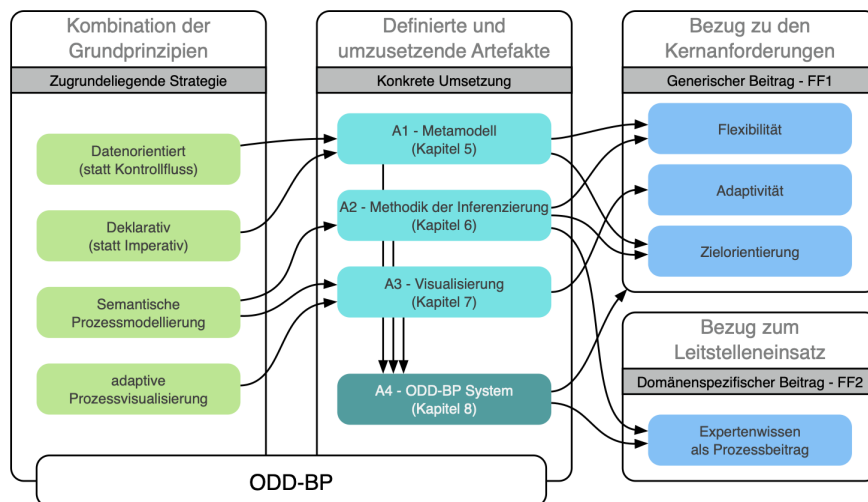


Abbildung 3.1: Übergeordneter Rahmen dieser Arbeit

Abbildung 3.1 zeigt den übergeordneten Rahmen dieser Arbeit, wonach die Kombination der Grundprinzipien die grundlegende Strategie des ODD-BP Ansatzes darstellt. Damit verbunden sind die einzelnen Artefakte (A1, A2, A3), welche in den nachfolgenden Kapitel 5, 6, 7 die konzipierte Methodik einführen. Die Ergebnisse dieser konzeptionellen Arbeit werden dann als viertes Artefakt (A4) in den Aufbau eines ODD-BP Systems in Kapitel 8 zusammengeführt und somit prototypisch ausführbar. Diese Anstrengungen dienen dem Zweck, den generischen Beitrag in Bezug auf Forschungsfrage 1 zu leisten, weshalb in der Grafik nochmals zurückgespiegelt wird, welche

Artefakte zur Erfüllung der einzelnen Kernanforderungen von FF1 beitragen.

Die Umsetzung des ODD-BP Systems mit einer einheitlichen Wissensbasis bietet dann noch weitergehende Möglichkeiten. So kann beispielsweise generalisiertes Expertenwissen innerhalb einer Domain-Ontologie in das semantische Prozessmodell eingebunden und dazu ausgenutzt werden, einen direkten Prozessbeitrag zu leisten. Diese Möglichkeit wird bei der Ausarbeitung des Systemaufbaus (Kapitel 8) und der Definition des experimentellen Aufbaus (Kapitel 9.2) aufgegriffen, um innerhalb der Evaluation die zweite Forschungsfrage (FF2) adressieren zu können. FF2 fragt: *„Kann in dem einheitlichen Ansatz generisches Wissen so integriert werden, dass dieses in einer Prozessausführung einen Beitrag leisten und die Anwender:innen unterstützen kann?“*

Im nachfolgenden Kapitel wird das für diese Arbeit ausgewählte Anwendungsszenario vorgestellt und als einen repräsentativen Vertreter von KiPs identifiziert.

4 Anwendungsszenario

Leitstellen und Lagezentren kommen im Falle eines persönlichen Notfalls eine zentrale Bedeutung zu, um schnell und zielgerichtet auf eine Situation reagieren, Schaden begrenzen und Leben retten zu können. In großen Leitstellen von Behörden und Organisationen mit Sicherheitsaufgaben (BOS) kommen entsprechende Meldungen über die Notrufnummer 112 bis zu 200 mal pro Stunde vor und markieren den Normalbetrieb. Das grundsätzliche Ziel besteht dabei in der effizienten Organisation der Einsatzkräfte mit einer zur jeweiligen Situation verhältnismäßigen Reaktion.

Im Falle von weitreichenden Ereignissen wie einer Pandemie, einem Hochwasser oder einem Großbrand müssen zudem mehrere Leitstellen (Rettungsdienst, Feuerwehr, Polizei) mit verschiedenen Standorten und regionalen Verantwortlichkeiten (über Kreis- oder Landesgrenzen hinweg) zusammenarbeiten. Gerade bei solchen Großereignissen liegt in einer gut koordinierten situationsspezifischen, intersektoralen und interorganisationalen Zusammenarbeit unterschiedlicher Leitstellen und Lagezentren die Chance, tiefgreifende Auswirkungen auf viele Wirtschafts- und Lebensbereiche zu verhindern oder zu mindern. Im Rahmen des Krisenmanagements müssen Maßnahmen zur Gefahrenabwehr mit deren Auswirkungen auf den öffentlichen und privaten Sektor, wie kritische Infrastruktur, systemische Liefernetzwerke, Wirtschaft und Industrie sowie Freizeitgestaltung, balanciert werden [107]. Daher sind neben den unmittelbar betroffenen Leitständen viele weitere Parteien auf eine funktionierende Informationsverteilung und Maßnahmenkoordination angewiesen.

Im Kontext der in Deutschland föderalen Verteilung von Zuständigkeiten haben sich ca. 250 BOS-Leitstellen etabliert, deren IT-Infrastruktur im „Masterplan Leitstelle 2020“ [126] als „Unikate“ und „Technologie-Inseln“ bezeichnet werden. Moderne Ansätze hinsichtlich einer leitstellenübergreifenden Datenplattform oder der Einsatz von KI-Methoden finden praktisch keine Anwendung, wie der Fachverband der Leitstellen in einem Positionspapier [90] feststellt. Gleichzeitig wird jedoch ein großer Bedarf konstatiert, der anhand von 14 Beispielen verdeutlicht wird. Eines dieser Beispiele adressiert direkt den in dieser Arbeit fokussierten Anwendungsfall einer Notrufbearbeitung.

4.1 Motivation

Im Kern liegt die Aufgabenstellung von Leitstellen darin, möglichst *schnell* die *richtigen* Entscheidungen zu treffen. Eine *schnelle* Entscheidung ist erforderlich, da es sich bei Notfällen oft um Situationen handelt, welche eine möglichst kurzfristige Reaktion erfordern. Eine *richtige* Entscheidung verlangt hingegen ein möglichst umfassendes Verständnis über eine Situation, was sich in einer detailreichen Datenrepräsentation widerspiegeln kann. Die Analyse und Interpretation solcher Daten nimmt Zeit in Anspruch, weshalb die beiden Attribute *schnell* und *richtig* oft schwer zu vereinen sind und einen angemessenen Kompromiss erforderlich macht.

Mit dem aktuellen Stand von Wissenschaft und Technik besteht die Chance, bei Notfällen und Krisensituationen mittels Datentechnologien und Künstlicher Intelligenz die Leitstellen bei dieser Aufgabe zu unterstützen. Beispielsweise können KI-basierte Dienste bei der Analyse und Interpretation umfangreicher Datenmengen helfen, indem wesentliche Aspekte eines Notfalls identifiziert oder bedeutende Zusammenhänge innerhalb der Daten geschlussfolgert werden können. Die Mitarbeiter:innen einer Leitstelle oder eines Krisenstabes können so vor einer Informationsflut bewahrt und in die Lage versetzt werden, schnell zu reagieren und die richtigen Entscheidungen zu treffen. Solche entscheidungsunterstützende Systeme [108] unterstreichen grundsätzlich die Aussicht auf substantielle Mehrwerte durch KI-basierte Dienste im Leitstellenumfeld.

Eine alltägliche Aufgabe aus dem Bereich der BOS-Leitstellen ist die täglich tausendfach vorkommende Abwicklung eines Notrufs über die Notrufnummer 112, welche sowohl in alltäglichen Notfällen als auch bei Großereignissen der erste Kontaktpunkt zwischen Leitstelle und Ereignissen darstellt. Nachfolgend wird die Notrufbearbeitung als Anwendungsszenario vorgestellt und als repräsentatives Beispiel eines wissensintensiven Prozesses eingeführt. Das dabei vorgestellte Beispiel wird fortlaufend in den nachfolgenden Kapiteln aufgegriffen und letztlich auch zur Evaluation verwendet. Das für diese Arbeit erforderliche fachliche Wissen zu dieser Domäne wurde von dem Team der Integrierten Leitstelle der Stadt Ludwigshafen (ILtS-Ludwigshafen) und dort auch mit der Unterstützung des ärztlichen Leiters eingebracht. Wenn nachfolgend auf domänenspezifisches Expertenwissen oder eine Experteneinschätzung Bezug genommen wird, dann referenziert dies die fachliche Expertise und die langjährige Erfahrung aus der ILtS-Ludwigshafen.

4.2 Der Anwendungsfall

Zur Bearbeitung eines Notrufs folgen die Mitarbeiter:innen der Leitstelle, nachfolgend als Disponent:innen bezeichnet, einem vordefinierten Schema, welches als standardisierte oder strukturierte Notrufabfrage (SNA) bezeichnet wird. Die Methodik der SNA zielt darauf ab, dass schnellstmöglich die wichtigsten Fragen gestellt werden, um den Anrufer lokalisieren und den Schweregrad des Notfalls einschätzen zu können. Im Falle einer vorliegenden lebensbedrohlichen Situation soll das Verfahren schnellstmöglich zu einer Disposition von Einsatzkräften und zu Ersthilfemaßnahmen wie Druckmassage oder Beatmung führen.

Die Abbildung 4.1 zeigt das initiale Basis-Segment der SNA-Methodik, wie es nach aktuellem Stand an der ILtS-Ludwigshafen praktiziert wird. Die mit einem Fragezeichen gekennzeichneten Rechtecke sind die Fragen der Disponenten. Je nach Antwort des Anrufers werden dann im weiteren Verlauf unterschiedliche, nachfolgende Fragen gestellt. Die roten, mit einem A gekennzeichneten Rechtecke sind Aktionen wie Hilfezusicherungen oder unmittelbare Maßnahmen.

Neben diesem Basis-Segment der SNA, das die wichtigsten Fragen für die lebensbedrohlichsten Notfallsituationen adressiert, gibt es noch vier Erweiterungen zur gezielten Abfrage von Sturzereignissen, Schlaganfall-Anzeichen, Geburtsvorfällen und Gefahrenmelder-Ereignissen. Insgesamt umfasst die SNA derzeit 50 Fragen, 12 mögliche Aktionen und eine Menge an Dispositionsoptionen; von einer ärztlichen Hotline über Rettungswagen und Notärzte bis hin zu dem Einsatz eines Rettungshubschraubers oder eines Feuerwehreinsatzes.

4.3 Die Anwender:innen

Im Kontext der SNA sind innerhalb der ILtS-Ludwigshafen vier unterschiedliche Rollen in den Ablauf integriert. Die für die ärztliche Leitung verantwortliche Person definiert allgemeine Einsatzrichtlinien, aufgrund derer festgelegt wird, wie abhängig vom jeweiligen Notfall und der jeweiligen Situation die Einsatzkräfte disponiert werden sollen. Eine schichtleitende Person hat für ein Zeitfenster von acht Stunden die Verantwortung für die Disponenten und koordiniert Zuständigkeiten und Pausenzeiten. Die Disponenten nehmen die Notrufe entgegen und versuchen schnellstmöglich, die zu dem jeweiligen Notfall geeigneten Maßnahmen und Dispositionen von Einsatzkräften vorzunehmen. Eine Person übernimmt dabei die Zuständigkeit

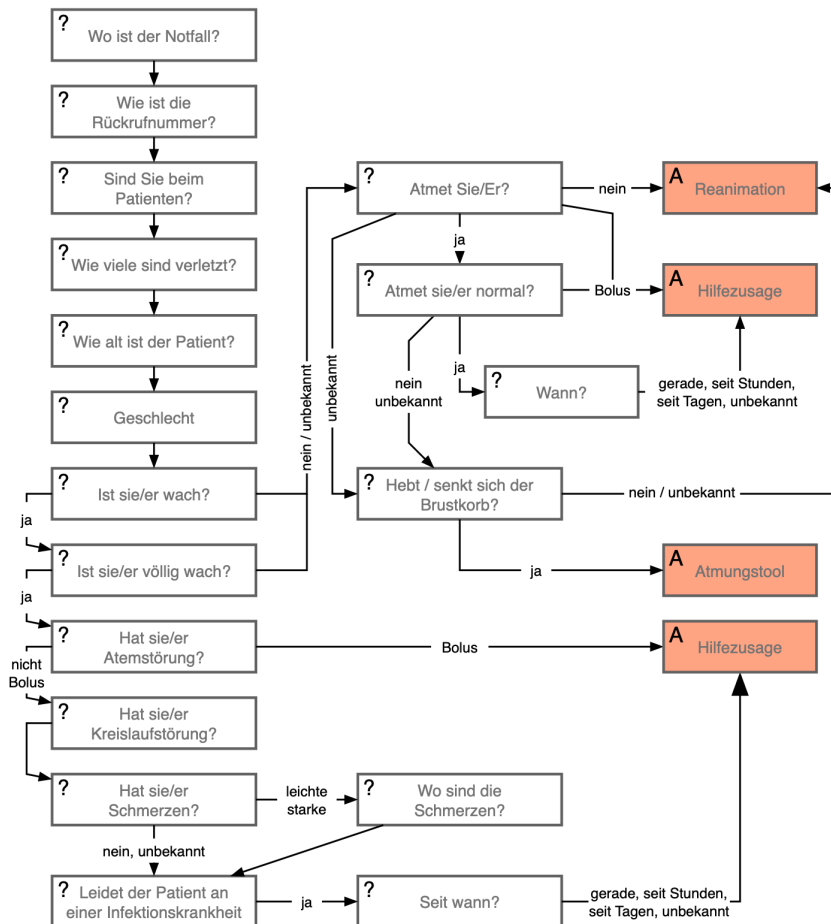


Abbildung 4.1: Basis-Segment der SNA-Methodik (ILtS-Ludwigshafen)

für den Funk und koordiniert die funkgestützte Kommunikation mit den Einsatzkräften. Dies dient der Entlastung der Disponenten, die im direkten Kontakt zu den anrufenden Personen stehen und erlaubt eine parallele Bearbeitung, indem Einsatzkräfte bereits alarmiert werden, während der Notruf noch im Gange ist.

In der ILtS-Ludwigshafen setzt sich das Team aus Vertretern des Rettungsdienstes und der Feuerwehr zusammen. Als Mindestanforderung wird dabei eine Ausbildung zum Rettungssanitäter vorausgesetzt und es wird im Rahmen einer sechswöchigen Schulung der Umgang mit dem Einsatzleitsystem sowie die Methodik der SNA geschult. An dieser Stelle darf erwähnt werden, dass insbesondere in den USA ein alternativer Ansatz verfolgt wird, bei dem die Mitarbeiter i.d.R. keine besonderen Voraussetzungen und Vorbildungen erfüllen müssen und die Kompetenz der Notrufabfrage einzig durch ein softwaregestütztes Verfahren geleistet wird. In Deutschland und in vielen anderen Ländern wird die Notrufabfrage hingegen wesentlich durch die Kompetenz der Disponenten getragen. Dies spiegelt sich auch in der Einschätzung der Experten der ILtS-Ludwigshafen wider, demnach der Ablauf der SNA in großem Maße von dem durch die Disponenten eingebrachten Wissen abhängig ist. In Folge dessen wirkt sich die Leitstellenerfahrung, aber auch die individuelle Tagesform des Disponenten wesentlich auf den Ablauf eines Notrufes und letztlich auf das Dispositionsergebnis aus.

4.4 SNA - Repräsentativ für KiPs

Nachfolgend wird eine Einordnung vorgenommen, inwieweit der vorgestellte Ablauf der SNA die in Kapitel 2.6 definierten grundlegenden Charakteristiken von KiPs aufzeigt und dementsprechend als ein repräsentatives Anwendungsszenario betrachtet werden kann.

1: Wissensgetrieben

Der Status und die Verfügbarkeit von Daten- und Wissensobjekten bestimmen die menschliche Entscheidungsfindung und beeinflussen direkt den Ablauf von Prozessaktionen und -ereignissen.

Die im Zuge der SNA erhobenen Daten sind der zentrale Ausgangspunkt für die von den Disponent:innen getroffenen Entscheidungen, sodass dieser Aspekt als voll zutreffend betrachtet werden.

2: Kollaborationsorientiert

Die Erstellung, Verwaltung und Ausführung von Prozessen erfolgt in einer kollaborativen Mehrbenutzer-Umgebung, in der menschenbezogenes und prozessbezogenes Wissen von und zwischen Prozessbeteiligten mit unterschiedlichen Rollen gemeinsam erstellt, geteilt und übertragen wird.

Im Kontext der Prozessausführung werden Beiträge von mehreren Beteiligten wie den ärztlichen Leitstellenleiter:innen, den Disponent:innen, Schichtführer:innen, funkverantwortlichen Personen und nicht zuletzt den Anrufer:innen geleistet. Zwischen diesen Personen findet teilweise eine eng verzahnte Kollaboration statt, sodass von einer Interaktion zwischen vielen unterschiedlichen Prozessteilnehmer:innen gesprochen werden kann.

3: Unvorhersehbar

Der genaue Aktivitäts-, Ereignis- und Wissensfluss hängt von situations- und kontextspezifischen Elementen ab, die möglicherweise nicht a priori bekannt sind, sich während der Prozessausführung ändern und über verschiedene Prozessfälle hinweg variieren können.

Da fast jeder Notruf von den Beteiligten als ein Unikat eingestuft wird und zu Beginn des Anrufs das Prozessergebnis vollständig unvorhersehbar ist, kann dieser Aspekt als voll zutreffend eingestuft werden.

4: Sich entwickelnd

Der tatsächliche Handlungsablauf ergibt sich allmählich während der Prozessausführung und wird Schritt für Schritt festgelegt, wenn mehr Informationen zur Verfügung stehen.

Auch dies muss als voll zutreffend betrachtet werden, da sich erst während des Notrufs sukzessive durch die erhobenen Daten der tatsächliche Handlungsbedarf abschätzen lässt.

5: Zielgetrieben

Der Prozess entwickelt sich durch eine Reihe von zu erreichenden Zielen, Zwischenzielen oder Meilensteinen.

Ein Notruf ist ein Prozess der gestartet wird ohne irgendwelche Vorkenntnisse und mit dem abstrakten Prozessziel, die Sicherheit einer unbekannt Person herzustellen. Erst mit der sukzessiven Erhebung von Daten kann das Prozessziel oder die Prozessziele konkretisiert und damit der SNA Prozess darauf ausgerichtet werden. Dies entspricht jedoch nicht dem etablierten

Umgang mit kontrollflussorientierten Ansätzen. Dabei ist jeder Prozess auf eines oder mehrere Prozessziele ausgerichtet und mit der Auswahl einer Prozessvorlage wird die Ausrichtung der Prozessinstanz festgelegt. Zur Auswahl einer Prozessvorlage stehen zum Beginn des Notrufs jedoch noch keine Informationen zur Verfügung. Zudem können sich aufgrund eines Notfalls eine beliebige Kombination unterschiedlicher Prozessziele ergeben. So könnten sich beispielsweise aus dem Vorfall eines Schiffsbrandes die drei Prozessziele, *Schutz einer Person mit Atemnot, Feuerbekämpfung* und *Warnung Schiffsverkehr* ableiten. Demzufolge kann auch dieser Aspekt als voll zutreffend bezeichnet werden.

6: Ereignisgetrieben

Der Prozessverlauf wird durch das Auftreten verschiedener Arten von Ereignissen beeinflusst, welche die Entscheidungsfindung der Wissensarbeiter beeinflussen.

Bedrohliche oder akute Notfall-Situationen sind oft agil und sind Teil einer sich permanent entwickelnden und verändernden Lage, innerhalb derer eintretende Ereignisse die Situation insgesamt beeinflussen und die Entscheidungen der Disponent:innen beeinflussen. Somit kann auch dieser Aspekt als voll zutreffend eingestuft werden.

7: Regelgetrieben

Prozessbeteiligte können von Einschränkungen und Regeln beeinflusst werden oder müssen diese einhalten, welche die Ausführung von Aktionen und die Entscheidungsfindung beeinflussen.

Die Richtlinien des ärztlichen Leiters einer Leitstelle legen fest, in welchen Notfällen und unter welchen Begleitumständen gewissen Rettungsmittel eingesetzt werden sollen, weshalb auch dieser Aspekt als voll zutreffend betrachtet werden kann.

8: Nicht wiederholbar

Die Schritte, die zur Bearbeitung eines bestimmten Falles oder einer bestimmten Situation unternommen werden, sind kaum wiederholbar, d. h. verschiedene Ausführungen eines Prozesses weichen voneinander ab.

Die Einzigartigkeit von Notfällen ist nicht nur ursächlich für einen unvorhersehbaren Verlauf, sondern ist auch der Grund dafür, dass Notrufe und

der damit verbundene Prozess der SNA nahezu keine Wiederholungen zulässt. Letztlich kann auch der letzte Aspekt als voll zutreffend bezeichnet werden.

Abgesehen von der nur teilweise zutreffenden Übereinstimmung hinsichtlich der Kollaboration sind alle anderen Aspekte der SNA in vollständigem Einklang mit den Charakteristiken von KiPs [32]. Weitere Beispiele für KiPs wurden im Rahmen des Semanas Forschungsprojektes untersucht und finden sich beispielsweise als Antragsverfahren im landwirtschaftlichen Bereich oder in der Bearbeitung von Baumängeln innerhalb der Baubranche.

4.5 Stand der Technik

Die grundsätzliche Methodik der SNA ist das Ergebnis jahrzehntelanger Erfahrung, welche von ursprünglich papiergebundenen Ablaufbeschreibungen und Checklisten inzwischen in die etablierten Leitstellen-Softwareprodukte integriert wurde. Die ILtS-Ludwigshafen arbeitet mit dem in Deutschland am weitesten verbreiteten System für nichtpolizeiliche BOS-Leitstellen (Behörden und Organisationen mit Sicherheitsaufgaben), der Software *Cobra* von dem Hersteller ISE GmbH, welche in ca 50% aller 250 BOS-Leitstellen eingesetzt wird. Die Methodik der SNA ist dabei in Form eines statischen Entscheidungsbaums implementiert. Nach der Beantwortung einer Frage wird entsprechend der in Abbildung 4.1 dargestellten Weise je nach Antwort die nächste Frage ausgewählt. Das System erlaubt dabei keine Abweichung und folgt im Grundsatz dem Vorbild eines kontrollflussorientierten Geschäftsprozesses. Damit unterliegt es auch allen Beschränkungen der in Kapitel 2.4 beschriebenen kontrollflussorientierten Ansätzen einschließlich der eingeschränkten Möglichkeiten hinsichtlich Flexibilität. Jede Form von Flexibilität muss nach dem Prinzip von „Flexibility by design“ (Kapitel 2.5) explizit modelliert werden und das zu jedem einzelnen Zeitpunkt der Prozessausführung. Überträgt man dies auf den der SNA zugrundeliegenden Entscheidungsbaum, dann müssten an jedem Knoten des Baumes alle möglichen Optionen einer Weiterführung des Notrufes explizit modelliert werden. Dem Anspruch an Flexibilität während der Prozessausführung der SNA kann dieser Ansatz somit nicht gerecht werden, weil ein unüberschaubar großes und nicht mehr wartbares Prozessmodell entstehen würde.

Neben dem Basis-Segment der SNA stehen noch vier spezialisierte Segmente zur Verfügung, welche jedoch bei Bedarf manuell ausgewählt und

gestartet werden müssen. Eine Integration in den Gesamtablauf oder eine automatische Erweiterung des Basis-Segments, abhängig von den bekannten Daten, findet nicht statt. Nach Rücksprache mit den Experten der ILtS-Ludwigshafen und auch mit Blick auf aktuelle Studien [90] unterscheiden sich die Software-Lösungen zwischen verschiedenen Anbietern lediglich in Details der umgesetzten Methodik, während alle die Abfragelogik nach dem Prinzip eines Entscheidungsbaums realisiert haben.

Im Fokus des Basis-Segment der SNA steht zunächst die Identifikation und Abwendung einer akut lebensbedrohlichen Situation mit Atem- oder Herz-Kreislauf-Stillstand. Da diese Ereignisse nur selten vorkommen und aufgrund der großen Vielfalt möglicher sonstiger Notrufursachen müssen laut Experteneinschätzung die Disponenten jedoch in mehr als 90% der Fälle von dem standardisierten Ablauf abweichen oder diesen für die jeweilige Situation um individuelle Fragen ergänzen. Diese Adaptionen erfolgen auf Grundlage des Ausbildungs- und Erfahrungsstandes jedes Einzelnen und führen i.d.R. zu einem oder mehreren vor-, ein- oder nachgelagerten Kommunikationsverläufen, welche außerhalb der SNA erfolgen und somit vom Einsatzleitsystem nicht unterstützt und erfasst werden. Es wurde auch davon berichtet, dass vereinzelt Notrufe vollständig außerhalb des Systems durchgeführt und diese dann nachträglich im System eingegeben wurden, um der vorgeschriebenen Dokumentationspflicht zu genügen. Darüber hinaus berichten die Disponenten, dass die Systematik der SNA sie in ihrer Arbeit und in ihren Entscheidungen oft ohne klare Regeln und Richtlinien zurücklässt, was zu Unsicherheiten führt. Im Ergebnis kann beobachtet werden, dass Einsatzkräfte oft umfangreicher disponieren, als dies nach objektiven Gesichtspunkten notwendig wäre. Was zunächst wie ein rein finanzieller Nachteil wirkt, wird auch zu einem qualitativen Nachteil, falls bei nachfolgenden Notfällen nicht auf wichtige Ressourcen zugegriffen werden kann, da diese gerade in einem anderen, weniger dramatischen, Fall gebunden sind. Im Ergebnis bietet die aktuelle systemische Umsetzung der SNA Methodik nur wenig Unterstützung abseits eines lebensbedrohlichen Notfalls, es werden vor-, ein- oder nachgelagerte Prozesssegmente außerhalb des Einsatzleitsystems durchgeführt, die Disponenten empfinden eine Unsicherheit in ihren Entscheidungen, was sich in einem nicht optimalen Prozessergebnis mit der Tendenz zu einer überhöhten Disposition widerspiegelt.

Diese Differenzen zwischen dem gelebten Prozess der Notrufabfrage und der dazugehörigen digitalen Repräsentation werden auch im Kontext anderer Studien zu KiPs [110] beschrieben. Als Ursache wird überwiegend die schlechte Eignung des kontrollflussorientierten Prinzips zur Ausführung von

KiPs angeführt, insbesondere aufgrund der Charakteristik, dass nicht Daten, sondern Aktivitäten im Mittelpunkt einer Prozessausführung stehen [67]. Dabei werden zudem die Fakten aus dem realen Vorgang meist nicht ausreichend innerhalb der Datenelemente der digitalen Repräsentation [71] abgebildet.

Mit Blick auf die zunehmende Digitalisierung unserer Gesellschaft kann davon ausgegangen werden, dass zukünftig noch wesentlich mehr Daten zur Verfügung stehen, welche in einen Prozesskontext eines Notrufes eingebunden werden sollten. Dies reicht von einfachen Metadaten wie der Telefonnummer des Anrufers bis hin zu Daten eines Fahrzeugs, dass per eCall, dem automatischen Notruf moderner PKW, im Falle eines Unfalls selbstständig den Notruf wählt und Daten wie die GPS-Koordinaten, Anzahl der Insassen, ausgelöste Airbags und Art des PKW-Antriebs (Benzin, Diesel, Gas, Elektro) übermittelt. Zudem stehen wir erst am Anfang moderner IoT-Geräte wie „Smart-Watches“, „Wearables“ und Systeme eines „Ambient Assisted Livings“, welche in der Zukunft eine immer stärkere Rolle als Datenquelle bei Notrufen spielen werden. Dies verstärkt die Notwendigkeit an eine Methodik zur Steuerung der SNA, welche mit den Daten aus unterschiedlichen Quellen und zu unterschiedlichen Zeitpunkten innerhalb eines Notrufprozesses umzugehen weiß. Aus dieser Perspektive muss von einem zunehmenden Auseinanderdriften, einer Divergenz, ausgegangen werden hinsichtlich der zukünftig steigenden Menge relevanter Daten zu einem vorliegenden Notfall und der Möglichkeit, solche Daten innerhalb der derzeitigen Methodik der SNA zielgerichtet nutzen zu können.

Vor diesem Hintergrund wurde in Gesprächen mit den Experten der ILtS-Ludwigshafen die aktuelle Systematik der SNA diskutiert und mit den dieser Arbeit zugrundeliegenden Kernanforderungen von KiPs verglichen. Davon ausgehend, dass das kontrollflussorientierte Prinzip des Entscheidungsbaums der SNA diesen Kernanforderungen nicht gerecht werden kann, sollten sich entsprechende Defizite als Bestätigung dieser Annahme finden lassen. Nachfolgend wird hierfür zunächst ein fiktiver Notruf als Beispiel eingeführt, anhand dessen anschließend die einzelnen Defizite der SNA zu den Kernanforderungen nachvollziehbar beschrieben werden können.

4.6 Fiktiver Notruf

Linksbündig und mit (D) gekennzeichnet wird der Dialogtext des Disponenten dargestellt. Die Antwort des Anrufers wird rechtsbündig und mit einem (A) gekennzeichnet. Die fortlaufende Nummerierung dient der späteren Referenzierung einzelner Fragen und Antworten im fiktiven Ablauf.

(D1) Wo ist der Ort des Notfalls?	Ich bin in der Schneegasse 12 in Ludwigshafen. (A1)
(D2) Unter welcher Nummer können wir Sie erreichen?	Unter 01234 56789 (A2)
(D3) Um was für einen Notfall handelt es sich?	Ich bin gestürzt, habe starke Schmerzen in meinem Knie. (A3)
(D4) Sie sind selbst betroffen?	Ja. (A4)
(D5) Sind noch weitere Personen verletzt?	Nein. (A5)
(D6) Wie alt ist die betroffene Person?	38 (A6)
(D7) Welches Geschlecht hat die betroffene Person?	weiblich (A7)
(D8) Ist die betroffene Person bei Bewusstsein?	Ja (A8)
(D9) Hat die betroffene Person Probleme mit der Atmung?	Nein (A9)
(D10) Hat die betroffene Person Probleme mit dem Kreislauf?	Nein (A10)
(D11) Hat die betroffene Person Schmerzen?	Ja (A11)
(D12) Wo sind die Schmerzen?	Mein Knie (A12)
(D13) Hat die betroffene Person eine Infektionskrankheit?	Nein (A13)
(D14) Kann jemand den Rettungskräften die Tür öffnen?	Ja (A14)
(D15) Ich schicke Ihnen einen Rettungswagen.	

Tabelle 4.2: Fiktiver Notruf: Sturz

Der Methodik der SNA folgend beginnt jeder Anruf mit der Abfrage des *Notfallortes* und einer *Rückrufnummer* unter der, im Falle eines Verbindungsabbruchs, die anrufende Person erreichbar ist. Im Anschluss wird nach der *Art des Notfalls* gefragt, woraufhin der Anrufer seine individuelle Situation beschreibt. Hierbei werden eine Menge an Daten übermittelt, welche zwar den Disponent:innen zur Einschätzung der Situation dienen, innerhalb des Systems aber nicht eingegeben werden können, da der zugrundeliegende Entscheidungsbaum dies nicht vorsieht und damit, ohne eine explizite Modellierung aller möglichen Notrufsituationen, nicht umgehen könnte. Wie aus der Abbildung 4.1 hervorgeht, ist die Frage nach der Art des Notfalls nicht Teil der SNA Methodik, es ist jedoch gelebter Alltag und fester Bestandteil jedes Notruf-Ablaufs, was sich im Rahmen der Evaluation bei allen Testfällen bestätigt hat. Aus diesem grundlegenden Problem, welches auf das kontrollflussorientierte Prinzip des Entscheidungsbaums zurückzuführen ist, erwachsen die nachfolgend dargestellten Defizite hinsichtlich der Kernanforderungen von KiPs.

4.7 Defizite

Der fiktive Ablauf eines Notfalls aus Tabelle 4.2 zeigt auf, wie bereits mit (A3) wesentliche Daten zur Notfall-Situation übermittelt werden. Der Entscheidungsbaum bietet keine Möglichkeit zur Eingabe dieser Daten zu diesem Zeitpunkt, da in Folge dessen alle möglichen weiteren Ausführungsoptionen abhängig von der Eingabe modelliert werden müssten. Faktisch müssen die Disponent:innen zunächst alle weiteren Fragen (D4) bis (D10) abarbeiten, bevor in (D11) erstmals die Daten aus (A3) teilweise eingegeben werden können.

Defizit Flexibilität

Der derzeitigen Systematik der SNA muss hinsichtlich der Kernanforderung an Flexibilität ein umfassendes Defizit zugeschrieben werden.

Da die Dateneingabe aus (A3), wie im beschriebenen Fall dargestellt, erst mit (D11) möglich wird, kann hinsichtlich Adaptivität und Zielorientierung keine direkte Untersuchung durchgeführt werden. Hierzu müsste durch die Ausnutzung von Flexibilität erst eine Situation entstehen, in der Adaptivität und Zielorientierung zum Tragen kommen können. Deshalb wird für die weitere Beurteilung angenommen, mit (A3) wären die entsprechenden Daten in das Prozessmodell der SNA eingeflossen.

Mit Erhalt der Erkenntnisse aus (A3) werden eine Reihe von nachfolgenden Fragen obsolet. So ist bekannt, dass die anrufende Person auch die betroffene Person (D4) ist. Weitere Fragen zum Bewusstsein (D8) oder der Atmung (D9) erscheinen in diesem Kontext wenig sinnvoll, während eine nicht vorhandene Frage zur Intensität der Schmerzen wesentlich relevanter erscheint. Im Rahmen des Entscheidungsbaumes der SNA sind keine Anpassungen an den kontrollflussorientierten Ablauf vorgesehen und der Ablauf bleibt auf den zuvor modellierten Umfang beschränkt.

Defizit Adaptivität

Der derzeitigen Systematik der SNA muss hinsichtlich der Kernanforderung an Adaptivität ein umfassendes Defizit zugeschrieben werden.

Ein Sturz mit starken Schmerzen im Knie, wie in (A3) übermittelt, kennzeichnet zunächst keine lebensbedrohliche Situation. Allerdings können in einer solchen Situation Anzeichen darauf hindeuten, dass die betroffene Person auch Merkmale eines Schockzustandes aufweist und die Gefahr eines Kreislaufzusammenbruchs besteht. Vor diesem Hintergrund werden neue und bislang im Ablauf nicht vorgesehene Fragen relevant, wie der Frage nach einer kalten fahlen Haut oder einem Schweißausbruch. Auch hier bietet der Entscheidungsbaum der SNA keine Möglichkeiten, den zuvor modellierten Ablauf zu verlassen und auf eine neue zur Situation passende Zielsetzung auszurichten.

Defizit Zielorientierung

Der derzeitigen Systematik der SNA muss hinsichtlich der Kernanforderung an eine Zielorientierung ein umfassendes Defizit zugeschrieben werden.

In Zusammenarbeit mit den Experten der ILtS-Ludwigshafen wurden die Defizite hinsichtlich Flexibilität, Adaptivität und Zielorientierung als die zentralen Probleme der etablierten Systematik der SNA identifiziert.

4.8 Fazit

Die Grundproblematik des eingeführten Anwendungsszenarios mit Blick auf die etablierte Systematik der SNA ist, dass mit einem statischen kontrollflussorientierten Ansatz eines Entscheidungsbaumes versucht wird eine hochdynamische Situation zu managen. Zu Beginn eines Notrufs stehen keine Informationen zur Verfügung, um ein zum Notfall passendes und auf ein entsprechendes Ziel hin optimiertes Prozessmodell zu wählen. Entsprechend wurde die bestehende Systematik der SNA darauf ausgerichtet, die besonders lebensbedrohlichen Fälle und ansonsten sehr abstrakt die restlichen Fälle abzubilden. Eine echte Spezialisierung der Abläufe hinsichtlich eines sich erst während des Notrufs abzeichnenden Prozesszieles ist im Rahmen eines imperativen und kontrollflussorientierten Ansatzes nicht möglich. Zudem können die Disponent:innen aufgrund des Zeitdrucks dies auch nicht manuell durch einen Wechsel in andere Prozessmodelle umsetzen. Eine gleichzeitige Ausrichtung auf mehrere, sich aus der Situation ableitende Prozessziele ist in diesem bestehenden Rahmen ebenfalls nicht vorstellbar. Dies spiegelt sich auch in den identifizierten Defiziten hinsichtlich der für KiPs relevanten Kernanforderungen wider.

Entsprechend wurden in Abstimmung mit den Experten der Leitstelle ILtS-Ludwigshafen die Kernanforderungen im Kontext des Anforderungsszenarios wie folgt formuliert. Die Disponent:innen benötigen:

- Flexibilität während der Datenerfassung, sodass alle relevanten Informationen eingegeben werden können, sobald diese anfallen und nicht abhängig davon, wann diese innerhalb eines Entscheidungsbaums adressiert werden. Dies begünstigt zudem die Integration weiterer externer Datenquellen wie eCall, Smartwatches und IoT-Geräte im Kontext des Notfalls.
- Adaptivität in der Prozessausführung und Prozessdarstellung, sodass die in der jeweiligen Situation wesentlichen Aspekte in den Fokus gerückt werden und unwichtige Aspekte ausgeblendet werden.
- Zielorientierung im Verlauf der Notrufbearbeitung, sodass ein Prozess unter unvollständiger Information über die Notrufsituation starten kann, um sich dann mit sukzessivem Eingang relevanter Daten auf die zur Situation geeigneten Prozessziele auszurichten.

Nachfolgend werden nun die grundlegenden Artefakte des auf KiPs ausgerichteten, neuen ODD-BP Ansatzes vorgestellt. Zur Evaluation des neuen Ansatzes wird dann das Anwendungsszenario nochmals aufgegriffen.

5 Konzeption des Metamodells

Entsprechend der in der Kapitel 3.2 dargestellten Strategie verfolgt diese Arbeit das Ziel der Kombination mehrerer Ansätze zu einem einheitlichen deklarativen, datenorientierten und semantisch beschriebenen Prozessmodell. Dieses Kapitel widmet sich dem ersten Artefakt, dem *Metamodell*, welches das grundsätzliche Sprachkonstrukt definiert, mittels dessen ein Prozess modelliert werden kann. Nachfolgend wird beschrieben, wie das Metamodell als eine Konzeptionalisierung Teil einer einheitlichen Wissensbasis wird. Dabei wird gezeigt, wie mithilfe dieser Formalisierung das Design von Prozessdefinitionen sowie die Ausführung von Prozessinstanzen festgelegt wird. Durch das Metamodell selbst wird dann ein deklarativer und datengetriebener Prozesscharakter geprägt, weshalb der Ansatz auch als datengetrieben bezeichnet werden kann. Nachfolgend wird die grundlegende Systemarchitektur, sowie das Metamodell von ODD-BP eingeführt, während Kapitel 6 die Methodik definiert, wie mit Hilfe der Inferenzierung wesentliche Fakten zur Prozessausführung gewonnen werden können.

5.1 Systemarchitektur

Der erste und wichtigste Unterschied zwischen etablierten BPM-Ansätzen und ODD-BP ist die semantische Integration allen prozessrelevanten Wissens in eine einheitliche Wissensbasis. In der Regel wirkt auf die Ausführung eines Prozesses Wissen aus verschiedenen Quellen und eingebettet in separierte Bereiche einer technischen Systemumsetzung ein. So enthält eine klassische Workflow-Engine das Wissen über die Modellierungs- und Ausführungsmöglichkeiten eines Prozesses in Form von Programmcode, welcher ein spezifisches Metamodell eines BPM-Ansatzes manifestiert. Das Datenmodell spiegelt Wissen zu einer Domäne wider und wird oft über das ER-Modell einer Datenbank definiert. Eine Prozessdefinition dient als Vorlage, wie eine bestimmte Art von Prozessen ausgeführt werden kann und wird über das Sprachkonstrukt des Metamodells beschrieben. Zusätzlich trägt jede Prozessinstanz Daten zusammen, die als digitaler Repräsentanz relevante Fakten über die reale Welt widerspiegeln. Wie in Abb.5.1 auf der

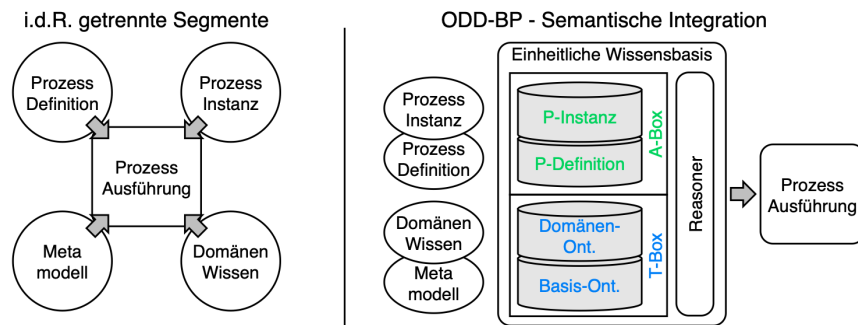


Abbildung 5.1: Prozessrelevantes Wissen und der Einfluss auf eine Prozessausführung

linken Seite dargestellt, sind diese Wissensquellen gewöhnlich in getrennten Segmenten eingebettet und jedes hat einen bedeutenden Einfluss auf die Ausführung eines Prozesses.

Im Gegensatz dazu folgt der ODD-BP Ansatz der Idee der SPM [37] und integriert diese Wissensteile semantisch in eine einheitliche Wissensbasis, wie in Abb.5.1 auf der rechten Seite dargestellt. Die Grundlage bildet eine Base-Ontologie, welche die Kernelemente eines Prozesses und die möglichen Beziehungen und Regeln definiert. Die Base-Ontologie selbst ist domänenunabhängig und wird in Kapitel 5.2 im Detail eingeführt. Die Domänenontologie definiert das Datenmodell für eine bestimmte Domäne sowie verallgemeinertes Domänenwissen mit wichtigem oder nützlichem Expertenwissen. Zusätzliche Ontologien können den Umfang des konzeptionellen Wissens noch erweitern um beispielsweise innerhalb einer Dokumentontologie allgemein gültiges Wissen zu Dokumenten und darin enthaltenen Daten zu formulieren. Die Ontologien definieren die Grundwahrheiten und bilden das konzeptionelle Fundament in der T-Box.

Eine Prozessdefinition und eine Prozessinstanz wird durch eine Reihe von verknüpften Individuen entsprechend den Konzepten und Beziehungen der Ontologien definiert. Diese Prozessmodelle werden zusammen mit dem situativen Wissen innerhalb der Prozessinstanzen in der A-Box als eine Menge von RDF-Statements im Stile von Subjekt, Prädikat, Objekt (entsprechend Kapitel 2.7) abgelegt.

Zusammen bilden T-Box und A-Box eine einheitliche Wissensbasis. Diese Wissensbasis bietet neben einer vollständigen und einheitlichen Wissensrepräsentation den Vorteil, dass eine Inferenzierungsmaschine (Reasoner)

prozessrelevante Fakten ableiten kann, um eine Prozessausführung zu unterstützen (siehe Kapitel 6).

5.2 Metamodell

Alle etablierten Workflow-Ansätze folgen einem spezifischen und vordefinierten Metamodell [96]. Wie in der Einleitung bereits aufgegriffen wird ein Metamodell definiert als [40, 77] „die Rahmen, Regeln, Einschränkungen, Modelle und Theorien, die für die Modellierung einer vordefinierten Klasse von Problemen anwendbar und nützlich sind.“ KiPs und ihre spezifischen Anforderungen [32] können als eine solche vordefinierte Klasse von Problemen betrachtet werden. Im Rahmen des ODD-BP-Ansatzes wird ein datenorientiertes Workflow-Metamodell definiert, das auf die Anforderungen von KiPs ausgerichtet ist. Im Vordergrund stehen dabei die im Kapitel 3.1 spezifizierten drei Kernanforderungen.

Das Kernstück von ODD-BP ist die Base-Ontologie, die alle Konzepte und Beziehungen definiert, um das Metamodell zur Definition und Ausführung von Prozessen aufzubauen. Die grundlegendsten Elemente eines Prozesses sind Aktivitäten, Datenobjekte und Dokumente und werden normalerweise in der einen oder anderen Weise in allen Workflow-Ansätzen berücksichtigt. Ihr spezifischer Charakter manifestiert sich im Metamodell durch die Art der Beziehungen zwischen diesen Prozesselementen. Dementsprechend wird der datenorientierte Charakter des ODD-BP Ansatzes durch die Beziehungen zwischen Aktivitäten auf der einen Seite und Dokumenten und Datenobjekten auf der anderen Seite erzeugt. Ein Ausschnitt der Base-Ontologie, die mit Hilfe der Web Ontology Language (OWL)[119, 22] erstellt wurde, ist in Abbildung 5.2 dargestellt, wobei Kreise mit gestrichelten Linien Klassen und gerichtete Kanten Beziehungen zwischen Klassen repräsentieren.

Verwendung englischer Bezeichner

Für eine präzisere und besser verständliche Beschreibung der nachfolgend ontologischen Konzepte, Beziehungen und Regeln wird folgende Regelung eingeführt. Begriffe wie Prozesse, Daten, Aktivitäten werden verwendet, um generelle Aspekte und Zusammenhänge aus dem Bereich von Geschäftsprozessen zu beschreiben. Da viele dieser Begriffe aber ebenso als Konzepte und Beziehungen im Zuge der Konzeptionalisierung verwendet werden, wird in diesem Fall der englische Bezeichner in kursiver Schreibweise - *Process*, *Dataobject*, *Task* - verwendet.

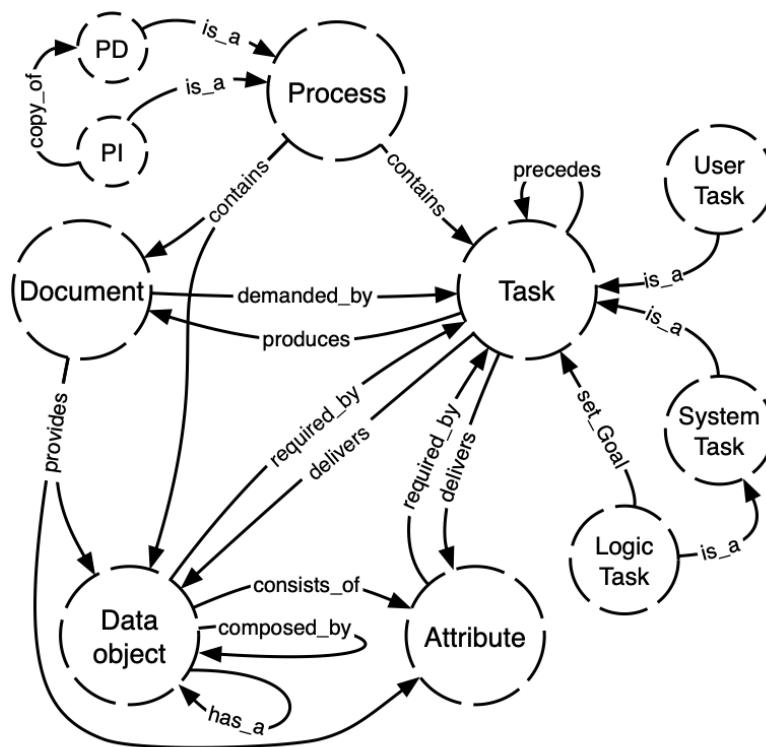


Abbildung 5.2: Auszug aus dem Metamodell der Base-Ontologie

Das Metamodell, wie in Abbildung 5.2 dargestellt, definiert primär, wie ein Prozess modelliert und ausgeführt werden kann. Den Ausgangspunkt bildet das Konzept *Process*, während die beiden Konzepte Prozessdefinition *PD* und Prozessinstanz *PI* als eine Spezialisierung von *Process* definiert sind. Durch die *contains* Beziehung wird festgelegt, welche unterschiedlichen Prozesselemente innerhalb eines Prozesses vorkommen können. Der Abbildung entsprechend kann demnach ein *Process* aus den bereits genannten grundlegenden Prozesselementen, aus Aktivitäten (*Task*), aus Datenobjekten (*Dataobject*) und aus Dokumenten (*Document*) aufgebaut werden.

Während das Metamodell in der Base-Ontologie das generelle Sprachkonstrukt für eine Prozessbeschreibung (innerhalb der T-Box) definiert, so werden einzelne Prozessdefinitionen und Prozessinstanzen (innerhalb der A-Box) durch eine Menge an Instanzen der entsprechenden Konzepte und ihre jeweiligen Verknüpfungen gemäß den vordefinierten Beziehungen festgelegt. Die Begriffe *Konzept* und *Beziehung* referenzieren die konzeptionelle Ebene (T-Box), während die Begriffe *Instanz* und *Verknüpfung* die realisierende Ebene (A-Box) adressieren.

Aktivitäten (*Tasks*) stehen im Zentrum jedes prozessführenden Systems und selbst bei einem datenzentrierten Ansatz ist eine zentrale Aufgabe die Identifikation ausführbarer Aktivitäten. Die Ausführbarkeit ist bei diesem Grundprinzip jedoch im Wesentlichen gekoppelt an die Verfügbarkeit der zur Ausführung der Aktivitäten benötigten Daten. Im Metamodell wird dies erkennbar durch die eingehenden Beziehungen zum Konzept *Task*. Über die *demanded_by* Beziehung wird ein *Document* verbunden, welche für die Ausführung eines entsprechenden *Tasks* erforderlich ist. Gleiches gilt für die *required_by* Beziehung zwischen *Dataobject* und *Attribute* zu *Task*. Im Gegenzug kann ein *Task* ein *Document* als Output produzieren (*produces*) oder aber auch ein *Dataobject* oder *Attribute* liefern (*delivers*). Die tiefere Bedeutung von *Dataobject* und *Attribute* wird im Kapitel 5.2.1 näher erläutert, aber die große Zahl an expliziten Beziehungen zwischen datentragenden Prozesselementen und einem *Task* kennzeichnet deutlich den datenorientierten Charakter des ODD-BP-Konzepts. Darüber hinaus kann in einer deklarativen Form auch eine kontrollflussähnliche *precedes* Beziehung zwischen *Tasks* modelliert werden.

Als eine Spezialisierung von *Task* erlauben die beiden Konzepte *User-Task* und *System-Task* die Unterscheidung in Aktivitäten, welche von einem Benutzer bzw. von dem System ausgeführt werden sollen. In Kapitel 5.2.2 wird das Metamodell um die Möglichkeit zur Definition und Zuordnung von Ressourcen erweitert. Als Spezialisierung von *System-Task* soll ein *Logic-Task*

die Möglichkeit bieten, vordefinierte Bedingungen zu überwachen. In Folge dessen können *Tasks* dann durch eine *set_Goal* Beziehung gezielt in einen Prozesszustand versetzt werden, der in Kapitel 6.4 im Detail eingeführt wird.

Dokumente stellen in einem Prozess zumeist wichtige Datenquellen dar, welche von außen eingebracht werden und eine Synchronisierung zwischen einem ablaufenden Vorgang und seiner digitalen Repräsentation erlauben oder welche von einem Prozess erzeugt und an externe Stellen versendet werden. Auch wenn im skizzierten Anwendungsszenario Dokumente nicht vorkommen, so spielen diese in vielen Prozessdefinitionen und Domänen eine wichtige Rolle. Mit Blick auf das Metamodell kann ein *Document* prozessrelevante *Dataobjects* bereitstellen (*provides*) und somit als ein Prozessinput betrachtet werden. Im Gegenzug können *Dataobjects* von einem *Task* benötigt (*required_by*) werden, um ein *Document* zu erzeugen (*produces*).

Datenobjekten kommt im Kontext eines datenorientierten Modellierungsansatzes naturgemäß eine besondere Rolle zu. Dabei können Datenobjekte untereinander in einem komplexen Beziehungsgeflecht stehen. Aus der generalisierten Perspektive eines Metamodells wird dazu lediglich in zwei grundsätzliche Beziehungen unterschieden. Mit der *has_a* Beziehung oder einer darauf aufbauenden Spezialisierung kann eine Beziehung zwischen *Dataobjects* ausgedrückt werden, welche zunächst keiner vorgegebenen semantischen Ausprägung folgt. Erst durch eine Spezialisierung wird eine jeweilige individuelle Bedeutung geformt. Als Beispiel kann man sich hierzu vorstellen, dass ein *Dataobject* „Person“ durch eine *has_a* Beziehung mit einer oder mehreren *Dataobjects* „Adresse“ verbunden ist. Alternativ könnte diese Beziehung auch über eine Spezialisierung zu *has_a* mit der Bezeichnung *residents_in* ausgedrückt werden und dadurch eine semantische Bedeutung erlangen. Mit der *composed_by* Beziehung wird eine taxonomische Grundbeziehung zwischen *Dataobjects* ausgedrückt, sodass entlang dieser Verbindung eine generische Aggregation denkbar ist. Beispielsweise könnte ein *Dataobject* „Baum“ durch die *composed_by* Beziehung oder einer darauf aufbauenden Spezialisierung mit *Dataobjects* wie „Stamm“, „Ast“ und „Blatt“ verbunden werden. Unter Ausnutzung dieser taxonomischen Grundbeziehung von *composed_by* können mehrere Elemente (Stamm, Äste, Blätter) im Zuge einer Aggregation zu einem gemeinsamen Element mit dem Titel (Baumelemente) zusammengefasst werden.

Darüber hinaus besteht (*consists_of*) ein *Dataobject* aus *Attributes*, ein Konzeptionsmerkmal das aufgrund seiner besonderen Bedeutung im nachfolgenden Kapitel 5.2.1 im Detail eingeführt wird.

5.2.1 Dataobjects und Attributes

Die generelle Datenorientierung des Metamodells lässt sich an den vielfältigen Beziehungen zwischen *Tasks* und den datenführenden Elementen *Dataobjects*, *Attributes* und *Documents* erkennen. Damit ähnelt die Herangehensweise von ODD-BP den Arbeiten zu artefaktzentrierten Ansätzen [17, 18, 28]. Durch die Modellierung der Input-Beziehungen zwischen *Dataobjects* und *Tasks* wird jedoch darüber hinaus über die Verfügbarkeit der Daten die Ausführbarkeit von Aktivitäten gesteuert.

Normalerweise organisiert ein Informationssystem Daten über die reale Welt mit Entitäten und Relationen. Mit Blick auf Datenbanken werden Entitäten als Einträge in einer Tabelle verwaltet, während innerhalb der eingeführten Wissensbasis Entitäten als Individuen einer bestimmten Klasse verwaltet. Ein solches Individuum repräsentiert ein Objekt der realen Welt, während seine Objektmerkmale als Eigenschaftswerte (data-properties) des Individuums gespeichert werden. Dieser Realisierung mangelt es an Aussagekraft, da eine Wissensbasis nach OWL2 (Kapitel: 2.7) keine Abhängigkeiten zwischen *Tasks* und Eigenschaftswerten (data-properties) ausdrücken kann und somit diese nicht zur Steuerung eines Prozesses verwendet werden können. Dies führt uns zu einer Konzeptionalisierung, in der die Objektmerkmale durch ein eigenes Konzept *Attribute* repräsentiert werden. Diese können als Schlüssel-Wert-Paare verstanden werden, wobei eine Instanz dieses Typs ein einzelnes Merkmal einer Entität repräsentiert. Die Existenz solcher *Attribute* Instanzen kann nun genutzt werden, um weitere Fakten über den Prozesszustand abzuleiten und damit einen Prozess zu steuern.

5.2.2 Tasks und Ressourcen

Tasks kennzeichnen in einem Prozess einen spezifischen Prozessbeitrag, der geleistet werden soll. Hierzu sind in der Regel weitere und zumeist externe Ressourcen erforderlich. Eine solche Ressource wird, wie in Abbildung 5.3 dargestellt, in der Konzeptionalisierung als *Actor* definiert. Ein *Process* kann beliebig viele *Actors* einbinden (*involves*).

Analog zur Unterscheidung in *User-Tasks* und *System-Tasks* wird auch ein *Actor* unterschieden in einen menschlichen und einen nicht-menschlichen *Actor*. Entsprechend werden als Spezialisierung zu *Actor* die Konzepte *User*, *Role* und *Agent* im Metamodell eingeführt. Ein *Agent* als das Konzept eines nicht-menschlichen *Actors* kann mit einem *System-Task* eine *performs*

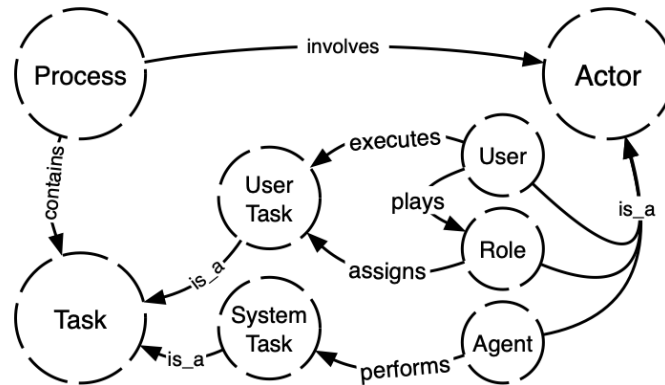


Abbildung 5.3: Auszug Task/Ressourcen Beziehungen aus der Base-Ontologie

Beziehung eingehen. Dadurch wird festgelegt, dass ein *System-Task* ausschließlich von diesem speziellen Ressourcentyp ausgeführt werden kann.

Anmerkung: Die Bezeichnung *Agent* wird hier nur als Vertreter einer abstrakten, nicht-menschlichen Entität verstanden, welche innerhalb eines *System-Tasks* einen Beitrag leistet. Dies kann im einfachsten Fall eine überprüfbare Bedingung, eine Funktion oder auch ein komplexer Dienst sein, der im Ergebnis den System-Task ausführt.

Da die Verfügbarkeit insbesondere bei Menschen als eine Ressource nicht immer gegeben ist (aufgrund von Arbeitszeiten, Urlaub, Krankheit), wird für *User-Tasks* ein Rollen-Konzept zur Modellierung umgesetzt. Innerhalb der Phase der Prozessplanung, manifestiert durch eine Prozessdefinition (PD), können einem *User-Task* eine oder mehrere *Roles* zugewiesen (*assigns*) werden. Ein *User* kann generell in einer oder mehreren *Roles* in einem Prozess mitwirken (*plays*). Wenn ein *User*, entsprechend der ihm zugeordneten *Roles*, innerhalb einer laufenden Prozessinstanz einen spezifischen *User-Task* ausführt, dann wird dies durch die *executes* Beziehung zum Ausdruck gebracht.

5.3 Prozessbeispiel unter Verwendung des Metamodells

Abbildung 5.4 zeigt einen Ausschnitt einer Prozessinstanz eines Notrufablauf, das mit der vorgestellten Base-Ontologie modelliert wurde. Die Formen mit durchgängigen Linien stehen für Instanzen eines bestimmten Typs. Dieser Typ (das dominierende zugrundeliegende Konzept) wird durch die Form entsprechend der dargestellten Legende repräsentiert.

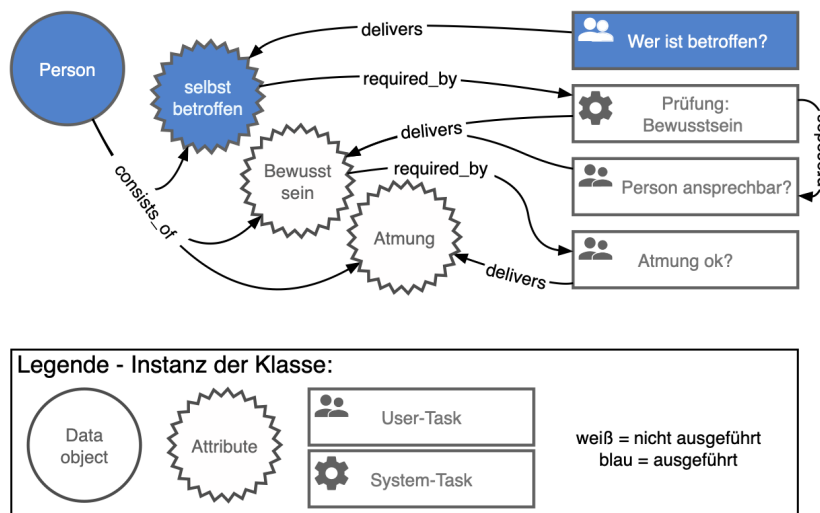


Abbildung 5.4: Prozessinstanz auf Basis des vorgestellten Metamodells

Die Prozessinstanz enthält ein *Dataobject*, das eine "Person" mit den drei *Attributes* "selbst betroffen", "Bewusstsein" und "Atmung" darstellt. Der *Task* "Wer ist betroffen?" liefert (*delivers*) als Output die Erkenntnis, ob der Anrufer "selbst betroffen" ist oder nicht. Dieses *Attribute* dient wiederum als Input für den *System-Task* "Prüfung: Bewusstsein" und den *User-Task* "Person ansprechbar?". Im nächsten Schritt wäre ein *Agent* (als synthetischer *Actor*) möglicherweise in der Lage, das *Attribute* "Bewusstsein" selbst zu bestimmen. Wenn der Anrufer selbst die betroffene Person ist kann von einem vorhandenen Bewusstsein ausgegangen werden. In diesem Fall kann auf den *User-Task* "Person ansprechbar?" verzichtet werden, da dieser *Task* keinen wertvollen Output liefern kann. Wenn der Anrufer nicht die betroffene Person ist, dann kann der *Agent* hinter "Prüfung: Bewusstsein" keine Aussage

zum Bewusstsein der betroffenen Person liefern und der *User-Task* "Person ansprechbar?" kann einen fehlenden Output liefern. Um sicherzustellen, dass zunächst der *Agent* seinen *System-Task* "Prüfung: Bewusstsein" durchführt, bevor einem *User* der *User-Task* "Person ansprechbar?" vorgeschlagen wird, wird eine *precedes* Beziehung zwischen diesen beiden *Tasks* definiert.

Prozesszustände werden im ODD-BP-Modell durch die Kennzeichnung nicht verfügbarer *Dataobject* und nicht ausgeführter *Tasks* als so genannte Platzhalter (*Placeholder*) dargestellt, ein zusätzliches Konzept, das im nächsten Kapitel ausführlich vorgestellt wird. Zu Beginn enthält eine Prozessinstanz nur Elemente mit *Placeholder*-Kennzeichnungen, die im Laufe der Ausführung entfernt werden, so dass der Prozess zunehmend aus Elementen besteht, die vorhandene (bedeutungsvolle) *Dataobjects* und ausgeführte *Tasks* beschreiben. Alle weißen Elemente des Auszugs sind als *Placeholder*-Elemente markiert, so dass nur das *Dataobject* "Person", das *Attribute* "selbst betroffen" sowie der *Task* "Wer ist betroffen?" bekannt bzw. bereits ausgeführt sind.

5.4 Fazit

Das eingeführte Metamodell definiert grundsätzlich einen deklarativen Workflow-Ansatz, bei dem Prozesse durch Abhängigkeiten und Beziehungen zwischen einzelnen Prozesselementen beschrieben werden. Der datenorientierte Charakter wird geprägt durch die vielfältigen Beziehungen zwischen datentragenden Elementen und *Tasks*. Ein datengetriebener Charakter ist bislang nur durch die Input-Beziehungen von datentragenden Elementen zu *Tasks* implizit angelegt, aber noch nicht realisiert. Im nachfolgenden Kapitel wird dieser datengetriebene Charakter durch explizite Regeln ausgeprägt und im Detail erläutert.

Innerhalb einer deklarativen Prozessmodellierung sind zumeist mehrere Aktivitäten ausführbar. Für eine leistungsfähige Prozessunterstützung ist somit nicht nur die Ausführbarkeit von *Tasks* alleine von Bedeutung, sondern es ist auch die generelle Relevanz ihrer Ausführung im Sinne eines Prozessbeitrages entscheidend. Hierfür werden im nachfolgenden Kapitel zusätzliche Regeln eingeführt, welche die Relevanz der Prozesselemente definieren.

6 Inferenzierung von Prozesszuständen

Dieses Kapitel widmet sich dem zweiten zentralen Artefakt, der Methodik und den Regeln zum Inferenzieren von Prozesszuständen für alle Prozesselemente einer Prozess-Instanz. Im Fokus stehen dabei vor allem drei Arten von Prozesszuständen, die *Ausführbarkeit* von *Tasks*, die *Ausführungsrelevanz* von *Tasks* und die *generelle Relevanz* beliebiger Prozesselemente zum Erreichen von Prozesszielen.

Das Inferenzieren auf einer Wissensbasis liefert grundsätzlich lediglich neue Fakten zu bestehenden Individuen und kennt keine Ausführungsrichtung einer zugrundeliegenden Prozessinstanz. Aus Sicht einer Prozessinstanz gibt es hingegen eine solche Richtung, ausgehend vom Zustand zum Prozess-Start hin zu einem finalen Zustand zum Prozess-Ende. Das Inferenzieren kann somit prozessrelevante Fakten liefern, welche sich entlang oder auch entgegen der Richtung in einem Prozessverlauf orientieren. Es ist erwähnenswert, dass auch ohne ein kontrollflussorientiertes Prinzip ein flexibler datenorientierter Ansatz wie ODD-BP immer noch eine Richtung im Graphen zusammen mit den Beziehungen zwischen den Prozesselementen hat. Abbildung 6.1 zeigt die drei Arten zu inferenzierender Prozesszustände mit Blick auf die Ausführungsrichtung einer Prozessinstanz.

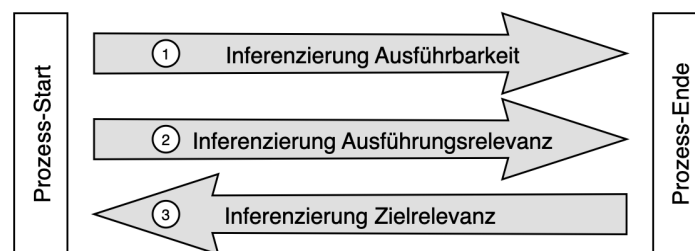


Abbildung 6.1: Inferenzierungskonzept

Entsprechend der Darstellung in Abbildung 6.1 werden zunächst in Ausführungsrichtung die Zustände der nicht ausgeführten Tasks aufgrund bekannter Datenabhängigkeiten inferenziert, um die Ausführbarkeit ① zu bestimmen. Die Grundidee dabei ist, dass ein Task als ausführbar betrachtet werden kann, wenn alle erforderlichen (*required_by*) *Dataobjects* und *Attributes* verfügbar sind.

Vergleichbar verhält es sich hinsichtlich der Ausführungsrelevanz ② eines *Tasks*. Ein Task wird als relevant betrachtet, falls seine Ausführung eine noch unbekannte Information liefert, also *Dataobjects* oder *Attributes* liefert, welche durch eine *delivers* Beziehung verknüpft sind.

Letztlich soll zudem für jedes Prozesselement seine Zielrelevanz ③ geschlossen werden. Hierzu wird entgegen der Ausführungsrichtung, von einem Prozessziel ausgehend, inferenziert, ob das jeweiligen Element zum Erreichen eines Prozessziels einen Beitrag leisten kann oder nicht.

Diese drei Teile des Inferenzmechanismus werden in den folgenden drei einzelnen Unterkapiteln vorgestellt. Zuvor benötigt die Base-Ontologie jedoch noch zusätzliche Konzepte, mit denen prozessrelevante Sachverhalte ausgedrückt werden können.

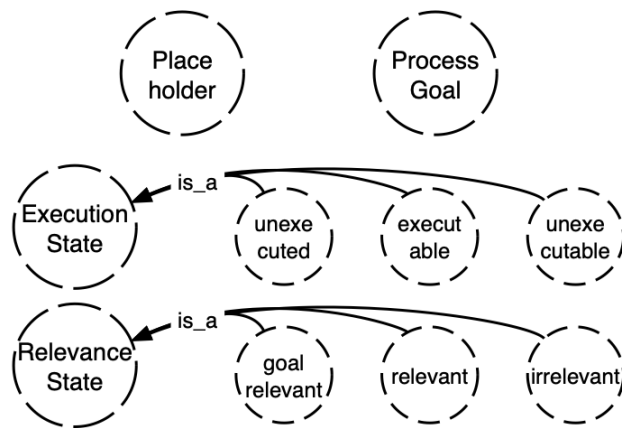


Abbildung 6.2: Erweiterung der Base-Ontologie

Abbildung 6.2 stellt eine Erweiterung der Base-Ontologie dar und führt als erstes das Konzept (*Placeholder*) ein, welches im Vorfeld bereits kurz erwähnt wurde. Jedes Individuum, welches eines der Prozesselemente (*Task*, *Document*, *Dataobject*, *Attribute*) repräsentiert, kann auch der Klasse *Placeholder* zugeordnet werden. Hierdurch wird gekennzeichnet, dass das entsprechende Element nur als Platzhalter zu interpretieren ist und noch keine Bedeutung hinsichtlich der Prozessinstanz repräsentiert. Bei der Erstellung einer neuen Prozessinstanz als eine Kopie einer Prozessdefinition wird jedes Prozess-Element auf diese Weise initialisiert. Sobald ein Element bedeutungsvoll wird oder ein *Task* ausgeführt wird, wird die *Placeholder*-Zuweisung des Individuums wieder entfernt.

Während das Konzept *Placeholder* verwendet wird, um den Ausführungszustand der Prozesselemente auszudrücken, ist das Konzept *Process Goal* dazu gedacht die Elemente zu kennzeichnen, welche im Prozessverlauf erreicht werden müssen. Dabei ist es unerheblich, ob die Zuordnung eines Individuums zu *Process Goal* explizit durch ein Statement erfolgt oder das Ergebnis einer Schlussfolgerung ist. Besonders die zweite Möglichkeit wird im späteren Verlauf (Kapitel 6.5) noch eine wesentliche Rolle bei der Integration und Anwendung von Expertenwissen spielen.

Um prozessrelevante Fakten abzuleiten, führen wir zusätzlich das Konzept *Execution State* (Ausführungsstatus) und *Relevance State* (Relevanzstatus) ein. Das Erste ist ausschließlich für *Task*-Elemente gültig und bietet drei Unterklassen (*unexecuted*, *executable*, *unexecutable*). Das Zweite bietet ebenfalls drei Unterklassen (*goal-relevant*, *relevant*, *irrelevant*) und kann grundsätzlich für beliebige Prozesselemente verwendet werden. Die tiefere Bedeutung der *State*-Konzepte wird durch die Formalisierung in den folgenden Abschnitten detailliert erläutert.

6.1 Inferenzierung der Ausführbarkeit

Unter den vielen Möglichkeiten des Inferenzierens durch OWL 2 ist für den angestrebten Zweck am interessantesten die Schlussfolgerung, dass ein Individuum zu einer bestimmten Klasse gehört; in unserem Fall die Schlussfolgerung, dass ein *Task* zur Klasse der (ausführbaren) *executable Tasks* gehört. Zur Präzisierung: Ein *executable Tasks* ist ein Individuum vom Typ *Tasks*, das zudem dem Konzept *executable* zugeordnet ist.

Gemäß Definition [22] ist eine Klasse in OWL 2 eine Menge von Individuen, die durch eine Spezifikation der Eigenschaften der Individuen beschrieben wird. Im Kontext einer datenzentrierten Geschäftsprozessmodellierung

haben *executable Tasks* die Eigenschaft, dass alle benötigten (*required_by*) *Dataobjects* und *Attributes* verfügbar sind [92]. Um auf die Ausführbarkeit schlussfolgern zu können, müssen wir dies in OWL 2 durch ein Konzept beschreiben. Da OWL 2 auf der Open-World-Assumption (OWA) (siehe Kapitel 2.7) beruht, ist dies eine Herausforderung, wenn die angestrebte Lösung praxistauglich sein soll. Die OWA macht es unmöglich zu sagen, ob etwas inexistent ist, bis seine Nichtexistenz festgestellt wurde [57]. Wenn sich also ein Inferenzierungsmechanismus für die Ausführbarkeit eines *Tasks* auf eine Klasse stützt, die durch eine Aussage über „Alles“ (Allquantor) definiert wird, muss im Kontext von KiPs ein erheblicher Aufwand in Kauf genommen werden. Dies kann an dem nachfolgenden Beispiel verdeutlicht werden. Beschreibt eine Klasse die Menge der *Tasks*, deren Input-Datenelemente *ALLE* verfügbar sind, so muss beim Schlussfolgern eine Aussage darüber getroffen werden, dass ein *Task* nicht mehr als die derzeit bekannten Input-Datenelemente hat bzw. haben könnte. Dies ist möglich, indem man Restriktionen (üblicherweise im englischen als *Closure* bezeichnet) hinzufügt, welche die Kardinalität der Beziehungen zwischen den Individuen einschränken [9]. In der Praxis bedeutet dies, dass für jede Variante einer solchen Kardinalität eine spezifische Version der Klasse *Task* definiert werden müsste. Da KiPs erhebliche Flexibilität zur Entwurfs- und Laufzeit [110] erfordern, müssen jedoch solche Datenabhängigkeiten und damit die Zuordnung zu einer jeweils anderen *Task*-Klassenvariante oft angepasst werden. In Folge dessen wäre dieser Ansatz unpraktisch, da beim Prozessdesign nicht nur Datenabhängigkeiten, sondern darüber hinaus auch erlaubte Kardinalitäten angepasst werden müssen.

Beschreibungslogiken, welche die Grundlage von OWL 2 bilden, erlauben alternativ das Schlussfolgern auf Grundlage des Existenzquantors. Beim Inferenzieren wird eine Aussage mit dem Existenzquantor „wahr“, wenn die Existenz von etwas bzw. die Gültigkeit eines Ausdrucks festgestellt wird. Sie bleibt „falsch“, wenn die Existenz bzw. Gültigkeit nicht bestätigt werden kann. Übertragen auf den diskutierten Anwendungsfall können wir das Gegenteil - die Unausführbarkeit einer Aufgabe - entscheiden, sobald ein einzelnes eingehendes benötigtes Datenelement als nicht verfügbar angegeben wird. Dies kann mit einem einzigen universellen *Task*-Konzept und einer generischen Regel realisiert werden, die keine *Closure*-Definition erfordert. Ein solcher alternativer Weg zur Beschreibung von ausführbaren *Tasks* durch die Untersuchung der Unausführbarkeit auf Basis der Datenverfügbarkeit wird im Folgenden vorgestellt und nutzt drei unterschiedliche Technologien, OWL2-Regeln, SWRL-Regeln und Abfragen (Queries). Während

OWL2-Regeln und SWRL-Regeln durch in Beschreibungslogik ausgedrückte Formeln definiert werden, werden die Queries durch SPARQL ausgedrückt. Auf diesem Wege wird ein letztlich notwendiges Closures für die Bestimmung der Ausführbarkeit auf einen späteren Zeitpunkt verlagert und mit Hilfe einer Abfrage umgesetzt. Die im nachfolgenden eingeführten Regeln verdeutlichen diese Vorgehensweise.

Im ersten Schritt definieren wir nicht ausgeführte *Tasks* als *Tasks*, die als Platzhalter markiert sind. Folglich entspricht die Klasse der nicht ausgeführten *Tasks* der Schnittmenge zwischen den Klassen *Placeholder* und der *Tasks* (R 1).

$$Task_{unexecuted} \equiv Task \sqcap Placeholder \quad (R\ 1)$$

Zur Vereinfachung der nachfolgenden Regeln, die sich teilweise in ähnlicher Form vielfach wiederholen würden, werden zwei Gruppierungen definiert. Entsprechend (R 2) wird *input* als eine Obermenge der *required_by* und *demanded_by* Beziehungen der Base-Ontologie definiert. Ebenso definiert die Regel (R 3) *Data* als das Superkonzept für die datentragenden Konzepte *Document*, *Dataobject* und *Attribute*. So kann nun mit Regel (R 4) verkürzt ausgedrückt werden, dass *Data*-Elemente die gleichzeitig eine *Placeholder*-Zuordnung besitzen der Klasse der *unavailable Data*-Elemente entsprechen. Da *unexecutable Tasks* mindestens ein *unavailable Data*-Element als *Input* oder mindestens ein *unexecuted Task* mit einer *precedes* Verknüpfung haben, kann die Zugehörigkeit zu dieser Klasse durch den Existenzquantor gemäß Regel (R 5) beschrieben werden.

$$input \sqsupseteq required_by \ \& \ input \sqsupseteq demanded_by \quad (R\ 2)$$

$$Data \equiv (Document \sqcup Dataobject \sqcup Attribute) \quad (R\ 3)$$

$$Data_{unavailable} \equiv Data \sqcap Placeholder \quad (R\ 4)$$

$$\begin{aligned}
Task_{unexecutable} \equiv & Task_{unexecuted} \sqcap \\
& (\exists input^-.Data_{unavailable} \sqcup \\
& \exists precedes^-.Task_{unexecuted})
\end{aligned} \tag{R 5}$$

Ein *Task* kann als *executable* betrachtet werden, wenn er zur Klasse der *unexecuted Tasks* gehört und nicht zur Klasse der *unexecutable Tasks* gehört. Anmerkung: Zur Vereinfachung werden bereits ausgeführte *Tasks* hierbei nicht betrachtet, da diese ihren jeweiligen Beitrag schon geleistet haben. Denkbar wäre aber natürlich auch, diese *Tasks* hinsichtlich ihrer Ausführbarkeit zu beurteilen und die Regeln leicht anzupassen.

Da wir keine Closure-Axiome einführen, werden die *executable Tasks* mit einer SPARQL-Query (R 6) abgeleitet. Tatsächlich kann die Ausführung und Interpretation der Query als eine Teilmenge der Wissensbasis betrachtet werden, die den Regeln der Closed-World-Annahme (CWA) folgt.

$$\begin{aligned}
& Task_{executable} \text{ (abbreviated as T_exec)} \\
& \text{SELECT ?T_exec WHERE \{ } \\
& \quad ?T_exec \text{ rdf:type :unexecuted} \\
& \quad \text{MINUS \{ ?T_exec rdf:type :unexecutable \}}
\end{aligned} \tag{R 6}$$

Obwohl diese alternative Beschreibung von *executable Tasks* recht umständlich erscheint, eignet sie sich sehr gut für den praktischen Einsatz, da Schnittmengen von Klassen einfach aus einer Wissensbasis abgefragt werden können. Da die genaue Abfrageroutine vom Anwendungsszenario abhängt, wird sie hier nicht weiter betrachtet. Stattdessen konzentrieren wir uns auf die Relevanz einer *Task*-Ausführung, die von der Existenz von Ausgabedaten abhängt und weitere Beiträge zu KiPs ermöglicht.

6.2 Inferenzierung der Ausführungsrelevanz

Bei der datenzentrierten Geschäftsprozessmodellierung können mehrere *Tasks* zum gleichen Ergebnis führen. Zum Beispiel könnte man entweder einen Anruf tätigen oder eine E-Mail schreiben, um dieselben Daten als Resultat zu erhalten. Wenn man einen Anruf tätigt und die Daten erhält, wird das Schreiben einer E-Mail irrelevant. Wenn der Anruf jedoch nicht

die gewünschten Daten liefert, kann man zusätzlich eine E-Mail schreiben. Daher sind *unexecuted Tasks*, deren Ausführung zu mindestens einem unbekanntem Datenelement führen würde oder die mindestens einem anderen *unexecuted Task* durch eine *precedes* Beziehung vorausgehen, *relevant* für eine Ausführung. Die Klasse solcher *relevant Tasks* wird mit der Regel (R 8) ausgedrückt. Beachten Sie, dass *output* die Obermenge der *delivers* und *produces* Beziehungen der Base-Ontologie (R 7) ist. Basierend auf unseren bisherigen Erkenntnissen können wir auch Kombinationen wie ausführbare relevante Aufgaben mit Hilfe einer weiteren SPARQL-Abfrage (R 9) beschreiben.

$$output \sqsupseteq delivers \quad \& \quad output \sqsupseteq produces \quad (R\ 7)$$

$$\begin{aligned} Task_{relevant} \quad \equiv \quad & Task_{unexecuted} \sqcap \\ & (\exists output.Data_{unavailable} \\ & \sqcup \exists precedes.Task_{unexecuted}) \end{aligned} \quad (R\ 8)$$

$$\begin{aligned} & Task_{executable_relevant} \text{ (abbreviated as } T_ex_re) \\ & \text{SELECT ?T_ex_re WHERE \{ } \\ & \quad ?T_ex_re \text{ rdf:type :unexecuted .} \\ & \quad ?T_ex_re \text{ rdf:type :relevant .} \\ & \text{MINUS \{ ?T_ex_re \text{ rdf:type :unexecutable \} \}} \end{aligned} \quad (R\ 9)$$

Ein auf den eingeführten Konzepten und Regeln basierender Inferenzierungsmechanismus betrachtet *Tasks* als relevant für die Ausführung, wenn ihre Ausgabe nicht bereits verfügbar und somit nützlich ist. Dabei bleibt jedoch unberücksichtigt, dass die Ausführung von *Tasks* obsolet werden kann, wenn die erzeugten Daten nicht zum Erreichen eines Prozessziels (*Process Goal*) benötigt werden, da der Mechanismus bislang nur den unmittelbar nachfolgenden Schritt berücksichtigt. Dies wird durch eine Erweiterung des obigen Beispiels verdeutlicht: Nehmen wir an, das Schreiben einer E-Mail oder das Tätigen eines Anrufs erfolgt, um Daten zu beschaffen, die von einer anderen Aufgabe benötigt werden, um ein Dokument zu erstellen. Das Dokument markiert dabei ein *Process Goal*. Wenn das Dokument bereits existiert, sind die *Tasks* des E-Mail-Schreibens oder des Telefonierens bei-

de obsolet, weil das *Process Goal*, für das sie ausgeführt werden würden, bereits erreicht ist. Da KiPs zielorientiert sind, da sie sich durch das Erreichen von Zwischenzielen oder Meilensteinen weiterentwickeln [32], würde der ODD-BP Ansatz davon profitieren, wenn die Zielrelevanz durch den Inferenzierungsmechanismus berücksichtigt würde.

6.3 Inferenzierung der Zielrelevanz

Im Folgenden betrachten wir Ziele als Individuen von *Tasks* und *Data-Elementen*, die der Klasse *Process Goal* zugeordnet sind, womit ausgedrückt wird, dass es einem besonderen Zweck dient diese auszuführen oder zu erzeugen. Die Besonderheit in diesem Fall ist, dass das Inferenzieren der *Goal-Relevanz* rückwärts - entgegen der Ausführungsrichtung des Prozesses - durch die Kette der Prozesselemente erfolgt. Dabei werden alle als *Placeholder* markierten Prozesselemente (abgekürzt PE) (R 10) (nicht nur *Tasks*) solange *goal-relevant*, bis das erste Element ohne *Placeholder* Zuordnung auftritt. Umgekehrt sind alle Elemente, die nicht in der Schleife enthalten sind, nicht *goal-relevant*, da ihre Ausführung oder Erzeugung nicht dazu beiträgt, ein unerfülltes *Process Goal* zu erreichen.

Damit sich der Inferenzierungsmechanismus wie oben skizziert verhält, benötigen wir einen Startpunkt, der offensichtlich ein als *Process Goal* gekennzeichnetes Prozesselement (PE) ist. Dies wird unser erstes *goal-relevantes* Prozesselement (in Formeln mit *goalRel* abgekürzt), wie es in Regel (R 11) definiert ist. Die Abhängigkeiten und die Selbstreferenzierung zwingt einen Reasoner zur rekursiven Inferenzierung der *goal-relevance* (gemäß Regel R 12), die sich ähnlich wie eine Schleife zur Bestimmung der *goal-relevance* für alle beitragenden Prozesselemente verhält. Da Schleifen eine klares Schleifenende benötigen, führt jeder *executed Task* oder jedes verfügbare *Data-Element* (nicht als *Placeholder* gekennzeichnet) zu einem Ende des rekursiven Inferenzierungsprozesses.

$$PE \equiv Data \sqcup Task \quad (R\ 10)$$

$$PE_{goalRel} \equiv Placeholder \sqcap ProcessGoal \quad (R\ 11)$$

$$\begin{aligned}
 PE_{goalRel} \equiv & Placeholder \sqcap (\\
 & \exists required_by. PE_{goalRel} \\
 & \sqcup \exists delivers. PE_{goalRel} \\
 & \sqcup \exists demanded_by. PE_{goalRel} \\
 & \sqcup \exists produces. PE_{goalRel} \\
 & \sqcup \exists provides. PE_{goalRel} \\
 & \sqcup \exists precedes. PE_{goalRel})
 \end{aligned}
 \tag{R 12}$$

Zum besseren Verständnis zeigt Abb.6.3 ein abstraktes Beispiel, das alle abgeleiteten Fakten (Ausführungszustände und Relevanzzustände) als Anmerkung neben jedem Prozesselement darstellt.

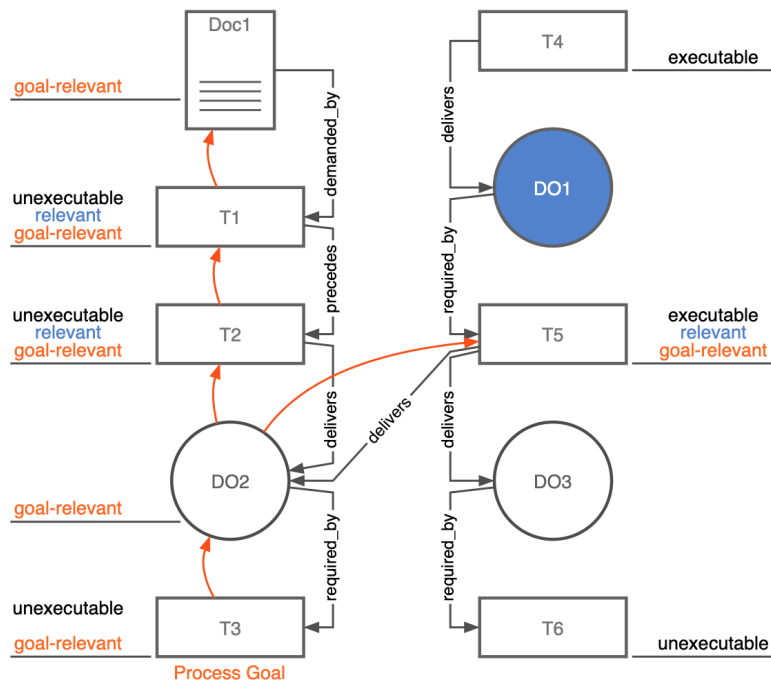


Abbildung 6.3: Beispiel mit *Execution-States* und *Relevance-States*

Demnach ist in Abbildung 6.3 *Task* T3 als *Process Goal* definiert, weshalb dieses Element gemäß R 11 auch als *goal relevant* betrachtet wird. Entsprechend der eingeführten Inferenzierungsregeln werden die *Tasks* T1-T6 in die Zustände *unexecutable*, *executable* und *relevant* eingestuft. Das Datenobjekt DO1 ist das einzige bedeutungstragende Element, während alle anderen Prozess-Elemente noch als *Placeholder* gekennzeichnet sind. T1, T2 und T5 sind die einzigen *Tasks*, die etwas Nützliches liefern könnten, also werden sie als *relevant* angesehen, während insbesondere T4 *irrelevant* ist, da DO1 bereits erfasst ist. T4 und T5 sind die einzigen *executable Tasks* in diesem Beispiel, da die erforderlichen *Dataobjects* vorhanden sind. Da T3 *goal-relevant* ist, beginnt an dieser Stelle eine Inferenzierungskette entgegen der Ausführungsrichtung. Folgt man den orangefarbenen Pfeilen rückwärts durch den Graphen, werden alle Prozess-Elemente ebenfalls *goal-relevant* (DO2, T2, T5, T1, Doc1). Die Inferenzierungskette wird erst bei dem bedeutungstragenden Element DO1 unterbrochen. Als Ergebnis werden DO1 und T4 nicht als *goal-relevant* eingestuft, ebenso wie DO3 und T6, die nicht zum einzigen *Process Goal* T3 beitragen. In Anbetracht dieses Ergebnisses ist die beste Wahl für einen Prozessfortschritt die Ausführung von T5, das *executable*, *relevant* und *goal-relevant* ist. Alternativ kann auch Doc1 in den Prozess aufgenommen werden, da dieses Element ebenfalls *goal-relevant* ist.

6.4 Logic-Task und Condition-States

Die vorgestellten Mechanismen sind ausreichend, um die Ausführbarkeit von *Tasks* anhand der Verfügbarkeit von Prozesselementen zu steuern. Allerdings haben wir bisher keine Möglichkeit, die Ausführbarkeit von *Tasks* anhand des Wertes eines bestimmten Prozesselements zu steuern. Hierzu können wir das Beispiel aus Abbildung 5.4 nochmals bemühen. Darin wird ausgedrückt, dass wir ein bedeutungstragendes (nicht *Placeholder*) *Attribut* [selbst betroffen] benötigen, um mit dem *Task* [Prüfung Bewusstsein] fortzufahren. Jedoch können wir nicht festlegen, ob der Wert des *Attributs* [selbst betroffen] dabei *wahr* oder *falsch* sein muss. Zu diesem Zweck erweitern wir unsere Base-Ontologie um *Logic Tasks* (*Condition*, *And*, *Or*, *Not*) und *Condition States* (*satisfied*, *unsatisfied*) wie in Abb. 6.4 dargestellt. Anmerkung: Nachfolgend werden zur Vereinfachung und zur Vereinheitlichung mit den Regeln die englischen Begriffen (Wert=value; wahr=true; falsch=false) verwendet.

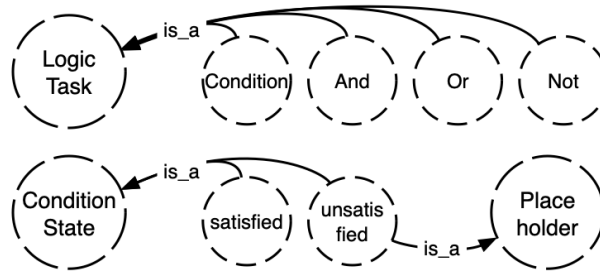


Abbildung 6.4: Logic Tasks extension of the Base-ontology

Von besonderer Bedeutung ist der neue Typ *Condition*, der eine Dateneigenschaft *Value* hat, genau die gleiche Dateneigenschaft, die auch für *Attributes* verwendet wird, um seinen jeweiligen Elementwert zu speichern. Als Vorbedingung muss eine *Condition* genau ein *Attribute* als Eingabe (*required_by*) haben. Wenn das *Attribute* eine *Placeholder* Einstufung hat oder der Wert beider Elemente ungleich ist, gilt die Bedingung als nicht erfüllt. Wenn der Wert beider Elemente gleich ist, gilt die Bedingung als erfüllt. Da dies die Ausdruckskraft von OWL2 übersteigt, erfordert es die Verwendung der Semantic Web Rule Language (SWRL), was uns zu den folgenden Regeln führt.

$$\begin{aligned} &Condition(?con) \wedge required_by(?att, ?con) \wedge \\ &Placeholder(?att) \rightarrow unsatisfied(?con) \end{aligned} \quad (R\ 13)$$

$$\begin{aligned} &Condition(?con) \wedge value(?con, ?v1) \wedge \\ &required_by(?att, ?con) \wedge value(?att, ?v2) \wedge \\ &unequal(?v1, ?v2) \rightarrow unsatisfied(?con) \end{aligned} \quad (R\ 14)$$

$$\begin{aligned} &Condition(?con) \wedge value(?con, ?v1) \wedge \\ &required_by(?att, ?con) \wedge value(?att, ?v2) \wedge \\ &equal(?v1, ?v2) \rightarrow satisfied(?con) \end{aligned} \quad (R\ 15)$$

Im Gegensatz zu anderen Prozesselementtypen wird eine *Condition* nie explizit als *Placeholder* eingestuft. Sobald eine *Condition* als *unsatisfied* in-

ferenziert wird, führt dies automatisch zu einer *Placeholder*-Zuweisung, da das Konzept *unsatisfied* ebenfalls eine Unterklasse des *Placeholder*-Konzepts ist (Abb. 6.4). Damit erfüllt eine *Condition* alle Voraussetzungen, um die zuvor vorgestellten Mechanismen zur Inferenzierung hinsichtlich Ausführbarkeit und Relevanzen zu unterstützen. Die anderen *Logic Tasks* (*And*, *Or*, *Not*) dienen der Kombination mehrerer Eingabelemente zur Erweiterung der allgemeinen Ausdrucksfähigkeit von Prozessdefinitionen und werden im Abschnitt 6.6 näher erläutert.

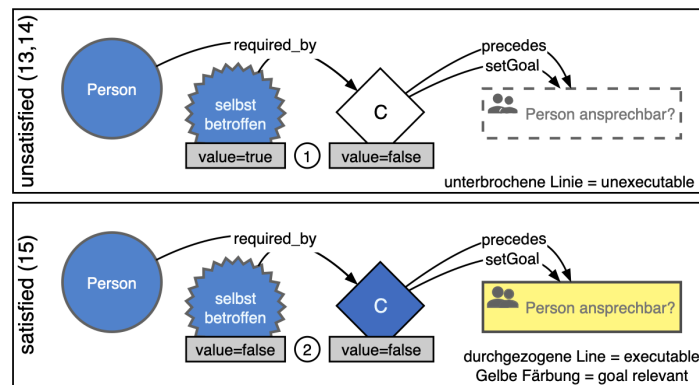


Abbildung 6.5: Beispiel mit einem *Condition* Element

Abb. 6.5 veranschaulicht die Verwendung eines *Condition*-Elements anhand eines Ausschnitts aus dem Beispiel aus Abbildung 5.4, der selbst betroffenen *Person*. Das raute förmige Element *C* ist ein *Condition* Element, das ein Eingabeattribut mit demselben Value-Typ voraussetzt. Sind die Werte unterschiedlich ①, ist die Bedingung nicht erfüllt und das *Condition* Element wird als *Placeholder* eingestuft. Sind die Werte gleich ②, ist die Bedingung erfüllt und das *Condition* Element wird als *executed* betrachtet. Durch die bereits bekannte Beziehung *precedes* kann der nachfolgende *Task* ausführbar (durchgezogene Linie) oder *unexecuted* (unterbrochene Linie) werden. Die Verknüpfung mit dem Typ *setGoal* wird im folgenden Abschnitt erläutert.

Anmerkung

Zur Vereinfachung erklären wir den Mechanismus nur an einem untypisierten Datenwert und unterscheiden nicht zwischen verschiedenen Wertetypen (wie String, Int, Date), was durch einen abstrakten Vergleichsausdruck (*unequal*, *equal*) in den Formeln (R 14, R 15) wiedergegeben wird.

6.5 Inferenzierung von Process Goals

Bisher haben wir *Process Goals* als eine explizite Zuordnung gesehen, indem wir einem Prozesselement das entsprechende Konzept zugewiesen haben. Dies könnte das Ergebnis eines vordefinierten Prozessziels sein, das in der Prozessdefinition angegeben und während der Instanziierung in eine Prozessinstanz kopiert wird. Alternativ könnte jedes Prozesselement von einem Benutzer während der Prozessausführung manuell als Prozessziel manuell definiert werden. Darüber hinaus ist es aber auch vorstellbar, dass ein Prozesselement in Folge einer Inferenzierung zu einem *Process Goals* eingestuft wird. Gerade der letzte Aspekt ist besonders interessant, da dies die Möglichkeit eröffnet die Zielsetzung eines Prozesses autonom, ohne Einwirkung eines Nutzers, an eine sich entwickelnde Datenlage und damit an eine sich verändernde Situation anzupassen. Dies erweitert die Ausdruckskraft unseres Metamodells und dient den spezifischen Anforderungen von KiPs an eine flexible und zielorientierte Prozessausführung [32].

Hierzu können wir das *Condition* Element, das wir in Kapitel 6.4 eingeführt haben, zusammen mit der Relation *setGoal* (Abb. 5.2) verwenden, um diese neue Funktionalität zu realisieren, die durch die Formel (R 16) definiert ist.

$$\begin{aligned} &Condition(?con) \wedge satisfied(?con) \wedge \\ &setGoal(?con, ?target) \rightarrow ProcessGoal(?target) \end{aligned} \quad (R\ 16)$$

Abbildung 6.5 hat bereits ein Beispiel für die Inferenzierung eines *Process Goal* dargestellt. Sobald die *Condition C* erfüllt ist, wird der *Task* [Person ansprechbar] nicht nur *executable*, sondern auch ein *Process Goal* und damit *goal relevant*, ausgedrückt durch eine gelbe Hintergrundfarbe. Zum besseren Verständnis wird der Mechanismus zum Ableiten von Prozesszielen im späteren Abschnitt 6.8 anhand eines weiteren Beispiels erläutert.

6.6 Herausforderungen durch die Open World Assumption

Die Implementierung eines Inferenzierungsmechanismus zur Realisierung der vorgestellten Funktionalität hat einige spezifische Anforderungen. Zunächst einmal müssen alle Formeln unter der Open-World-Annahme (OWA) entscheidbar sein. Dies ist der Fall, da die Formeln (1 - 15) gültige Ausdrücke innerhalb der Sprache OWL 2 DL sind oder einfache disjunkte Klassenbeziehungen darstellen, die durch SPARQL-Abfragen handhabbar sind. Die Formeln (16 - 19) sind gültige SWRL DL-sichere Regeln. Dies bedeutet, dass alle vorgestellten Regeln Teil des entscheidbaren Fragments der Prädikatenlogik erster Ordnung sind.

Die Open-World-Assumption führt zu weitreichende Einschränkungen hinsichtlich der Inferenzierbarkeit von Zusammenhängen und die Einführung disjunktiver Klassen (R 6, R 9) ist eine Möglichkeit, diese Einschränkungen in einzelnen Fällen zu überwinden. So kann mit der Fokussierung auf die Nicht-Ausführbarkeit von *Tasks* und den damit im Zusammenhang stehenden Bedingungen einfacher umgegangen werden als mit der Ausführbarkeit von *Tasks*, was das Inferenzieren von Prozesszuständen erlaubt, ohne ein Closure unterschiedlicher Task-Typen einführen zu müssen.

Beim Umgang mit den *Logic-Tasks* (*And*, *Or*, *Not*) wird es jedoch sehr mühsam, die OWA-Einschränkungen zu überwinden. Zum besseren Verständnis erläutern wir das Problem mit dem *Or Logic-Task*. Das gewünschte Verhalten eines *Or*-Prozesselements ist, dass wenn mindestens eines von vielleicht vielen Eingabeelementen bedeutungsvoll ist (nicht *Placeholder*), das *Or*-Element ebenfalls bedeutungsvoll sein sollte. Das heißt, wir müssen das *Or*-Element der Klasse *Placeholder* zuordnen, wenn **alle** Eingabeelemente ebenfalls *Placeholder* sind. Dies geht mit der Herausforderung einher zu wissen, wann der Zustand **alle** erreicht ist.

Eine Alternative dieses Problem zu überwinden besteht darin, mehrere *OR*-Unterklassen zu definieren, die alle eine feste Anzahl von Eingabeelementen haben, um die Annahme von OWA zu schließen, dass es weitere (noch unbestimmte) Eingabe-Elemente geben könnte. In Kombination mit spezialisierten SWRL-Regeln für jede dieser *OR*-Unterklassen könnte die Inferenzierung einer *Placeholder*-Zugehörigkeit technisch hergeleitet werden. Im Ergebnis müsste der Prozess-Designer je nach Anzahl der Input-Elemente das richtige *OR*-Prozesselement wählen. Da der Entwurfsprozess einer Prozessdefinition durch kontinuierliche Änderungen gekennzeichnet ist, erscheint dies kein praxistauglicher Weg zu sein, weshalb dies extern, ohne Inferenzierer, mit Hilfe von Agenten umgesetzt wird. Alternativ könnte auch mit SPAQRL-Abfragen gearbeitet werden, welche inhärent ein Closure durchführen. Hierzu müssten dann die Ergebnisse wieder in Form von Statements in die Wissensbasis zurückgespiegelt werden, damit diese als Basis für weitere Inferenzierungsschritte zur Verfügung stehen.

6.7 Prozessunterstützung mittels Expertenwissen

Die bisher diskutierten Beispiele folgen einer Prozessdefinition auf Basis der durch die Base-Ontologie vorgegebenen Konzepte, Relationen, OWL2- und SWRL-Regeln. Als Erweiterung dieser Methodik kann auch verallgemeinertes Expertenwissen, eingebettet durch weitere Regeln in der Domänenontologie, verwendet werden, um eine Prozessausführung zu beeinflussen. Eine solche Regel könnte ein bestimmtes Prozesselement als *Process Goal* definieren, sobald bestimmte Vorbedingungen erfüllt sind.

Zur Verdeutlichung wird nachfolgend eine Regel definiert, welche sich ausschließlich auf Instanzen eines spezifischen Konzepts auswirkt. Hierzu wird in der Domain-Ontologie das Konzept „*Cause of Unconsciousness*“ (CUC) (deutsch: „Ursache der Bewusstlosigkeit“) eingeführt, das eine Spezialisierung zu *Dataobject* darstellt. Durch die Regel R 17 wird jede Instanz dieses Typs als *Process Goal* definiert, sobald derselbe Prozess eine Instanz des

Attributs „conscious“ mit einem Wert „no“ enthält.

$$\begin{aligned}
 & Process(?proc) \wedge CUC(?cuc) \wedge conscious(?con) \wedge \\
 & contains(?proc, ?cuc) \wedge contains(?proc, ?con) \wedge \\
 & valueStr(?con, ?val) \wedge swrlb : matches(?val, "no") \\
 & \rightarrow ProcessGoal(?cuc)
 \end{aligned}
 \tag{R 17}$$

Sobald die Zugehörigkeit eines CUC-Elements zur Klasse *Process Goal* geschlussfolgert wird, greifen die bereits etablierten Mechanismen, um die *GoalRelevance* (R 11, R 12) auf verbundene Prozesselemente zu übertragen. Mit anderen Worten: Sobald die Daten innerhalb eines Prozesse aufzeigen, dass wir es mit einer bewusstlosen Person zu tun haben, richtet sich der Prozess auf ein neues zusätzliches Prozessziel aus um mehr über die Ursache der Bewusstlosigkeit herauszufinden. Dieses Beispiel wird im folgenden Abschnitt nochmals aufgegriffen und näher erläutert.

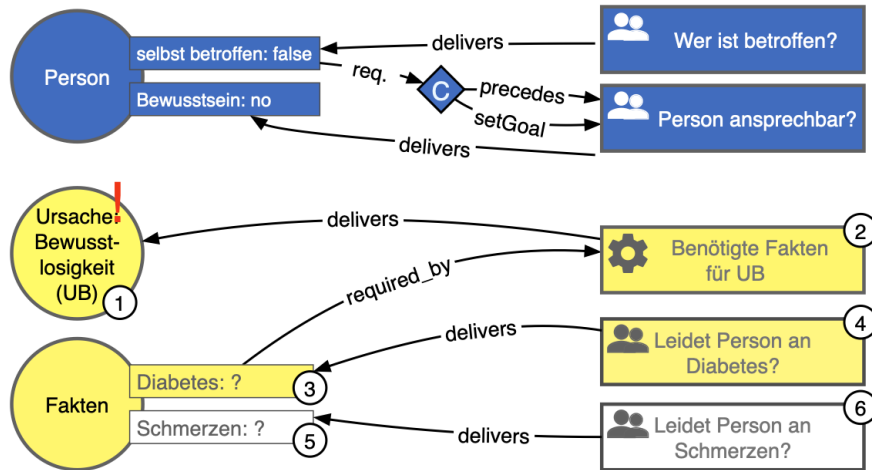
6.8 Demonstration der Inferenzierungsmethodik

Der vorgeschlagene Inferenzierungsmechanismus ermöglicht die Verwendung von Ontologien, um nicht ausgeführte *Tasks* in datengesteuerten Prozessen in den Dimensionen Ausführbarkeit und Relevanz zu klassifizieren, während *Process Goals* explizit festgelegt oder auf Grundlage von Expertenwissen geschlussfolgert werden können. Da die Relevanz durch den Zustand der zu liefernden Datenelemente und der Prozessziele bestimmt wird, können Benutzer *Tasks* auswählen, basierend auf dem Maß, in dem ihre jeweilige Ausführung zu einem Prozessfortschritt beitragen würde.

Die Fähigkeiten des Inferenzierungsmechanismus werden im Folgenden anhand von Protégé 5.2.0¹ und dem Ontologie-Reasoner Pellet² gezeigt. Die Demonstration verwendet einen erweiterten Auszug aus dem Notrufverfahren, wie in Abbildung 6.6 (Seite 109) gezeigt. Zur besseren Lesbarkeit werden die Attribute als Rechtecke dargestellt, die mit dem zugehörigen Datenobjekt verbunden sind, einschließlich des Wertes ihrer Dateneigenschaft

¹Protégé ist ein frei verwendbarer Ontologie-Editor der Stanford University, online verfügbar unter: <https://protege.stanford.edu>

²Pellet ist unter Doppellizenz freigegeben und online verfügbar unter: <https://github.com/stardog-union/pellet>



<p>Ursache Bewusstlosigkeit ①</p> <p>Definiert als:</p> <ul style="list-style-type: none"> - Dataobject - CUC - Placeholder <p>Inferenziert zu:</p> <ul style="list-style-type: none"> - ProcessGoal (17) - Goal relevant (11) 	<p>Diabetes ③</p> <p>Definiert als:</p> <ul style="list-style-type: none"> - Attribute - Placeholder <p>Inferenziert zu:</p> <ul style="list-style-type: none"> - Goal relevant (12) 	<p>Schmerzen ⑤</p> <p>Definiert als:</p> <ul style="list-style-type: none"> - Attribute - Placeholder <p>Inferenziert zu:</p>
<p>Benötigte Fakten für UB ②</p> <p>Definiert als:</p> <ul style="list-style-type: none"> - SystemTask - Placeholder <p>Inferenziert zu:</p> <ul style="list-style-type: none"> - Goal relevant (12) - Unexecutable (5) 	<p>Leidet Person an Diabetes? ④</p> <p>Definiert als:</p> <ul style="list-style-type: none"> - UserTask - Placeholder <p>Inferenziert zu:</p> <ul style="list-style-type: none"> - Executable (6) - Relevant (8) - Goal relevant (12) 	<p>Leidet Person an Schmerzen? ⑥</p> <p>Definiert als:</p> <ul style="list-style-type: none"> - UserTask - Placeholder <p>Inferenziert zu:</p> <ul style="list-style-type: none"> - Executable (6) - Relevant (8)

Abbildung 6.6: Beispiel mit Darstellung der Inferenzierungsmechanismen

valueStr. Zudem werden nur die wichtigsten Beziehungen dargestellt, um eine reduzierte Sicht auf die wichtigsten Aspekte dieses Beispiels zu ermöglichen.

Der obere Teil von Abbildung 6.6 stellt eine Prozessinstanz dar, die teilweise bereits ausgeführt ist (blau gefüllte Elemente). Die gelb hervorgehobenen Elemente sind allesamt mit der höchsten Relevanz (*goal relevant*) ausgestattet, während die weißen Elemente nur von einer geringeren Relevanz sind. Der untere Teil der Abbildung zeigt die zugehörigen Individuen innerhalb der Wissensbasis mit ihren explizit zugeordneten (definiert als) Klassen. Zusätzlich werden die abgeleiteten (inferenziert zu) Klassen dargestellt, inklusive eines Verweises auf die jeweils verwendeten Regeln.

Der ausgeführte Teil der Prozessinstanz (blau) liefert die Fakten, dass der Anrufer nicht *selbst betroffen* und die betroffene Person nicht bei *Bewusstsein* ist. Ohne direkten Bezug zum Element ① und mit Hilfe von Regel (20) erlaubt das in der Domänenontologie eingebettete Expertenwissen, dieses Element als *ProcessGoal* abzuleiten, was es aufgrund von Regel (14) zudem *goal-relevant* macht. Mit Hilfe der definierten OWL2- und SWRL-Regeln können die Zustände weiterer Prozesselemente abgeleitet werden. Die Systemaufgabe ② ist bisher *unexecutable* (R7), wird aber *goal relevant* (R15), da ihr Beitrag ein anderes zielrelevantes Element ① unterstützen würde. Das Gleiche gilt für das Attribut *Diabetes* ③, das ebenfalls als *goal relevant* (R15) identifiziert wird. Alle *Tasks*, die das Attribut ③ liefern können, werden ebenfalls *goal relevant* und mit Blick auf *Task* ④ lassen sich noch weitere Zustände (*executable* (R8) und *relevant* (R10)) ableiten. Im Gegensatz dazu ist das Attribut *Schmerzen* ⑤ kein Input für ② und wird somit nicht als zielrelevant angesehen. Dementsprechend wird auch die Aufgabe ⑥ nicht zielrelevant und wird lediglich als *executable* (R8) und *relevant* (R10) identifiziert. Ihre geringer bewertete Relevanz ergibt sich daraus, dass sie etwas Unbekanntes ⑤ liefern kann, aber nicht *goal relevant* ist. Dieses kleine Beispiel stellt einen Fall dar, in dem sich eine Prozessinstanz aufgrund der bekannten Daten an ein neues Prozessziel anpasst, was zu einer Neuberechnung der Relevanz für alle Prozesselemente führt.

6.9 Fazit

Die eingeführten Regeln definieren gemeinsam eine Methodik zum Inferenzieren von Prozesszuständen. Mit jeder Veränderung von Prozessdaten wird unter Anwendung dieser Regeln eine Prozessplanungsprozedur durchlaufen, woraufhin die Prozesszustände aller Prozesselemente neu festgestellt werden. Damit bildet die Inferenzierungsmethodik mit dem Verfahren einer Prozessplanungsprozedur eine zentrale Grundlage für den angestrebten flexiblen, adaptiven und zielorientierten ODD-BP Ansatz und repräsentiert gleichzeitig einen fundamentalen Unterschied zu den sich sequentiell entwickelnden Prozesszuständen kontrollflussorientierter Verfahren. Gleichzeitig kann mit diesem Ansatz auf eine klassische Workflow-Engine verzichtet werden und das Metamodell und die Regeln werden integraler Bestandteil einer einheitlichen Wissensbasis.

Mit den eingeführten Verfahren zur Bestimmung der Ausführbarkeit und der Relevanz von Prozesselementen wird die Grundlage für eine leistungsfähige Prozessunterstützung geschaffen. Letztlich können Anwender:innen das Potential eines solchen Ansatzes jedoch nur ausschöpfen, wenn die gewonnenen Freiheitsgrade nicht durch eine unübersichtliche Darstellung aller möglichen Ausführungsoptionen verloren gehen und das prozessführende System hilfreiche und klare Empfehlungen für den nächsten Prozessschritt vorschlägt. Das nachfolgende Kapitel knüpft exakt an diesem Punkt an und beschreibt ein Verfahren für eine adaptive Prozessvisualisierung (aPV).

7 Adaptive Prozessvisualisierung

Dieses Kapitel fokussiert eine adaptive Prozessvisualisierung (aPV), das dritte zentrale Artefakt, welches die Anwender:innen bei ihrer Arbeit mit Prozessen durch eine geeignete Darstellung bestmöglich unterstützt. Der Ausgangspunkt ist hierbei ein Prozess, welcher vollständig mit allen Prozesselementen visualisiert werden kann. Die grundlegende Hypothese hierbei ist, dass ein Verfahren existiert, welches durch die Adaption der Prozessvisualisierung die Aussagekraft erhöht. Dies beinhaltet zunächst ein Adaptionsverfahren, welches eine bestehende Prozessvisualisierung (PV) als Input erhält und eine adaptierte PV (aPV) zurückliefert. Die Erwartungshaltung ist nun, dass die Aussagekraft(aPV) größer ist als die Aussagekraft (PV).

Im Vordergrund stehen dabei die Anwender:innen mit ihren individuellen Anforderungen und Bedürfnissen, welche abhängig sind von der jeweiligen Rolle, innerhalb derer sie am Prozess mitwirken. Im Wesentlichen kann zwischen drei unterschiedlichen Gruppen an Prozessteilnehmern unterschieden werden.

- *Prozess-Designer:innen* widmen sich dem grundsätzlichen Aufbau und der Struktur eines Prozesses, in dem eine Prozessdefinition modelliert wird, welche als Vorlage für spätere Prozessinstanzen genutzt werden kann.
- *Prozessmanager:innen* fokussieren vor allem den generellen Fortschritt von Prozessinstanzen und überwachen, dass ihre Durchführung nicht ins Stocken gerät.
- *Prozessanwender:innen* wirken bei der Durchführung von Prozessinstanzen mit, erbringen einen spezifischen Beitrag und tragen dadurch einen Teil zur Erreichung der Prozessergebnisse bei.

Während *Prozess-Designer:innen* zum Modellieren vor allem eine gute Gesamtübersicht über alle Prozesselemente einer Prozessdefinition sowie deren Beziehungen untereinander benötigen, verlangen *Prozessmanager:innen* vor allem nach einer möglichst kompakten und übersichtlichen Darstellung des Prozesszustandes einer Prozessinstanz. In dieser Arbeit und speziell in diesem Kapitel werden hingegen die besonderen Bedürfnisse von *Prozessanwender:innen* (nachfolgend verkürzt als Anwender:innen bezeichnet) fokussiert, da sie von einer verbesserten Prozessunterstützung (entsprechend Kapitel 2.9) durch eine aussagekräftigere aPV profitieren können.

Betrachtet man bisherige Arbeiten zum Thema Prozessvisualisierung [88, 95, 20, 103], so berücksichtigen diese vor allem die Anforderungen von Prozess-Designern, da diese Personengruppe eine komplexe und anspruchsvolle Aufgabe zu erfüllen hat und die Qualität des Ergebnisses wesentlich durch die dabei verwendeten Werkzeuge verbessert werden kann. Die Bedürfnisse der Anwender:innen werden seltener untersucht [66], da diese durch das dominierende kontrollflussorientierte Prinzip in der Regel keine oder nur wenig Ausführungsoptionen innerhalb eines spezifischen Prozesszustandes haben. Es wird deshalb häufig davon ausgegangen, dass Nutzende lediglich durch eine Taskliste anstehende Aktivitäten eines oder mehrerer Prozesse angezeigt und zur Ausführung angeboten bekommen. Dass dies speziell mit Blick auf KiPs unzureichend ist, wurde in einigen Arbeiten adressiert [43, 91] und dabei hervorgehoben, dass der gesamte Kontext eines wissensintensiven Prozesses für die Durchführung einer einzelnen Prozessaktivität von Bedeutung ist.

Mit Blick auf das in dieser Arbeit verfolgte Ziel einer flexiblen Prozessausführung von KiPs kommt noch eine weitere Motivation hinzu. Anwender:innen sollen dabei unterstützt werden, möglichst frei und flexible einen Prozess ausführen und dabei aus einer Vielzahl von Optionen auswählen zu können oder auch ungebunden eine direkte Dateneingabe durchführen zu können. Hierzu wird eine Darstellung erforderlich, welche weit über die einfache jedoch begrenzte Anzeige einer Taskliste hinausgeht. Gleichzeitig erscheint es wenig vorteilhaft, wenn Anwender:innen zu jedem Zeitpunkt alle Aktivitäten und Datenelemente in vollem Detailgrad angezeigt werden, da dies in einem komplexen Prozess zu einer unübersichtlichen Darstellung führt, was eine Prozessausführung nicht unterstützen, sondern sogar behindern könnte. Ziel muss es also sein, eine breite Auswahl von vielversprechenden Ausführungsoptionen und Zugriff auf die wesentlichsten Datenelemente einer Prozessinstanz in einem spezifischen Prozesszustand zu bieten.

Dies rückt die Frage in den Vordergrund: „Was sind die Merkmale einer vielversprechenden Ausführungsoption bzw. wie wird ein wesentliches Datenelement identifizierbar?“ Einen Ausgangspunkt bietet die im vorherigen Kapitel entwickelte Methodik zur Inferenzierung von Prozesszuständen einzelner Prozesselemente; insbesondere die Erkenntnis zur Ausführbarkeit, Ausführungsrelevanz und Zielrelevanz erscheint hier ein guter Ansatzpunkt zu sein.

In diesem Kapitel wird zunächst das generelle Verfahren einer Prozessvisualisierung beschrieben. Nachfolgend wird der Transfer einer semantisch beschriebenen Prozessdefinition innerhalb der Wissensbasis hin zu einzelnen graphischen Prozesselementen unterschiedlicher Typen definiert. Im Anschluss werden generelle Methoden zur Modifikation einer Prozessvisualisierung eingeführt, sowie ein Verfahren zur Adaption der Prozessdarstellung beschrieben. Grundlage hierzu bildet eine Methodik zur Berechnung eines individuellen und kontextbezogenen Relevanzwertes für jedes Prozesselement. Zur Steuerung des Verfahrens wird eine Parametrisierung als Teil der semantischen Prozessbeschreibung eingeführt und dessen Verwendung anhand eines nachfolgenden Beispiels verdeutlicht.

7.1 Verfahren der Prozessvisualisierung

Ein Prozess besteht aus einer Menge an Daten, welche im Rahmen dieser Arbeit durch eine Menge von Statements innerhalb der Wissensbasis hinterlegt sind. Von diesem Punkt ausgehend müssen mehrere Verfahrensschritte durchlaufen werden, um zu einer adaptiven Prozessvisualisierung zu gelangen.

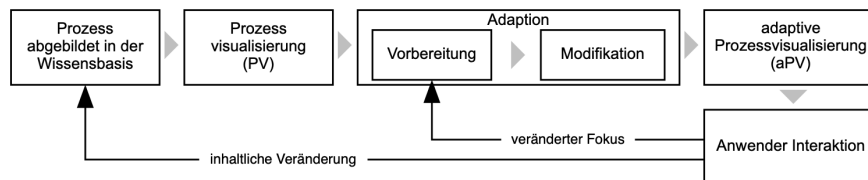


Abbildung 7.1: Verfahren der Prozessvisualisierung und Interaktion

Abbildung 7.1 zeigt das grundlegende Verfahren der aPV mit einzelnen Verfahrensschritten und bezieht in einer abstrakten Form auch die Einwirkung der Anwender:innen mit ein. Ausgangspunkt ist der Prozess, wie er durch die Menge an Statements innerhalb der Wissensbasis persistiert ist. In einem ersten Schritt wird hieraus eine originäre und allgemeine Prozessvisualisierung erstellt. Anschließend erfolgt die eigentliche Adaption für den jeweiligen Nutzenden, ein Verfahrensschritt der sich grob in die Vorbereitung und in die Modifikation der PV unterteilt. Das Ergebnis, eine adaptive anwenderbezogene Prozessvisualisierung wird nun den Anwender:innen präsentiert. Durch die Interaktion der Anwender:innen mit der aPV können grundsätzlich zwei Arten von Veränderungen eintreten. Eine inhaltliche Veränderung durch Ausführung von Aktivitäten oder der Ergänzung von Daten wirkt sich bis in die Wissensbasis aus und führt zu einem erneuten Durchlaufen des geschilderten Verfahrens. Alternativ können Anwender:innen auch lediglich ihren Fokus auf ein anderes Prozesselement richten (anklicken innerhalb der aPV) was keine inhaltliche Veränderung innerhalb der Wissensbasis zur Folge hat, aber zu einer veränderten Adaption der PV führen kann.

Die nachfolgenden Kapitel greifen die einzelnen Aspekte der Verfahrensschritte auf und erläutern im Detail die zugrundeliegenden Methoden und Verfahren.

7.2 Grundlagen einer Prozessvisualisierung

Wie in Kapitel 5.1 beschrieben besteht eine Prozessinstanz aus einer Menge an Statements innerhalb der eingeführten einheitlichen Wissensbasis. Eine Prozessvisualisierung übersetzt den in der Wissensbasis persistierten Prozesszustand in eine graphische Darstellung. In vorherigen Abbildungen wurde bereits vereinzelt eine graphische Notation mit expliziten Legenden verwendet, um spezifische Sachverhalte näher zu erläutern. Nachfolgend wird eine graphische Notation der Prozesselemente einer PV und der Zusammenhang zur zugrundeliegenden Repräsentation innerhalb der Wissensbasis grundlegend eingeführt und in den nachfolgenden Kapiteln entsprechend weiter verwendet.

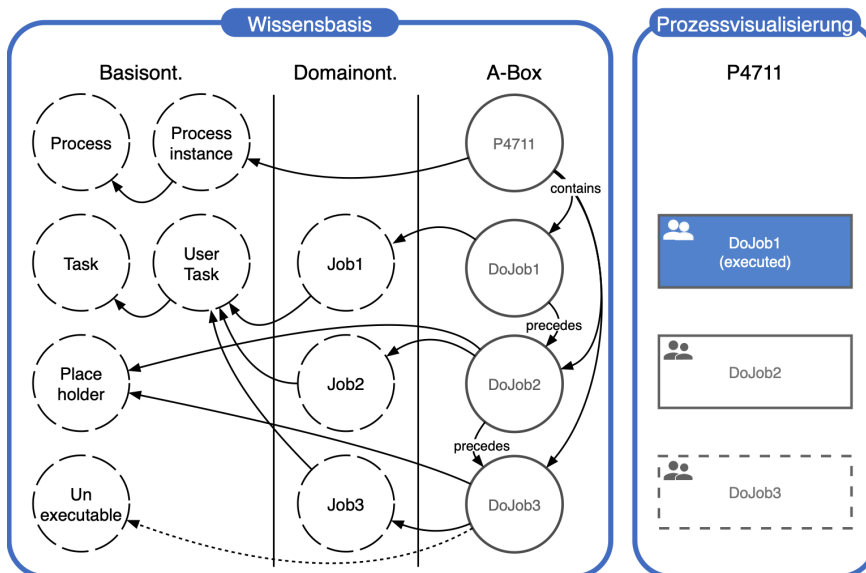


Abbildung 7.2: Transfer von drei Task-Prozesselementen in eine PV

Abbildung 7.2 zeigt einen Prozessausschnitt mit drei Prozesselementen aus zwei unterschiedlichen Perspektiven. Die linke Box zeigt die Repräsentation innerhalb der Wissensbasis mit den einzelnen Individuen und ihren Beziehungen zur zugrundeliegenden Konzeptionalisierung an, während die rechte Box die sich daraus resultierende Prozessvisualisierung der drei Prozesselemente darstellt.

Innerhalb der Wissensbasis werden durch Kreise mit einer durchgezogenen Linie die Individuen innerhalb der A-Box dargestellt, welche zusammen eine Prozessinstanz mit exakt drei Tasks formen. Dies wird definiert durch das Element P4711, welches über eine *contains*-Verknüpfung drei Prozesselemente an sich bindet. Die zugrundeliegenden Konzepte innerhalb der Basis- und der Domain-Ontologie werden als Kreise mit einer unterbrochenen Linie repräsentiert. Alle durchgezogenen Linien ohne explizite Beschriftung stellen eine einfache *is_a* Beziehung dar.

Das Individuum *DoJob1* ist demnach vom Typ *Job1*, was wiederum vom Typ *UserTask* ist, welches selbst wiederum eine Spezialisierung vom Typ *Task* ist. Entsprechend der getroffenen Festlegung ist *DoJob1* bereits ausgeführt, da keine Verbindung zum Konzept *Placeholder* existiert. Innerhalb einer Prozessvisualisierung wird dieses Individuum mit all seinen existierenden Beziehungen in Form eines einfachen blau ausgefüllten (=ausgeführt) Rechtecks abgebildet. Das Symbol in der linken oberen Ecke repräsentiert dabei den Task-Typ.

Ähnlich verhält es sich mit dem Individuum *DoJob2*, welches allerdings dem Konzept *Placeholder* zugewiesen ist und somit bislang noch nicht ausgeführt wurde. Durch die *precedes* Verknüpfung von *DoJob1* zu *DoJob2* wird eine Ausführungsbeschränkung vorgegeben. Da diese jedoch erfüllt ist, kann das Prozesselement als nicht ausgeführt aber ausführbar betrachtet werden, was innerhalb der PV durch ein nicht ausgefülltes Rechteck (= nicht ausgeführt) mit durchgezogener Linie (=ausführbar) dargestellt wird.

Das dritte Individuum *DoJob3* steht ebenfalls in einer *precedes* Beziehung, allerdings bezieht sich die Ausführungsbeschränkung auf das Prozesselement *DoJob2*, welches selbst noch nicht ausgeführt worden ist. Entsprechend der eingeführten Regeln kann somit zu *DoJob3* der Prozesszustand *Unexecutable* geschlussfolgert werden, was durch die gepunktete Linie innerhalb der Wissensbasis zum Ausdruck gebracht wird. Transferiert in die PV wird dies durch ein nicht ausgefülltes Rechteck mit gestrichelter Linie (=nicht ausführbar) dargestellt.

Durch die Kombination von Form, Füllung und Rahmenlinie kann so mit nur drei graphischen Elementen innerhalb der PV die Essenz von 13 Elementen der Wissensbasis zum Ausdruck gebracht werden.

In gleicher Weise verhält es sich mit Prozesselementen vom Typ Document. Da es zu diesem Typ keine Ausführbarkeit zu Inferenzieren gibt, wird lediglich in zwei grundlegende Zustände unterschieden, der Verfügbarkeit eines Dokuments und der nicht Verfügbarkeit, was wieder durch eine *Placeholder*-Relation festgelegt wird. Entsprechend wird in Abbildung 7.3 eine Form zur Repräsentanz von Dokumenten eingeführt. *Überweisung ABCD* steht entsprechend der bekannten Regelung für ein existierendes und bekanntes Dokument, während das Prozesselement *Überweisung* angibt, dass ein Dokument von entsprechendem Typ noch erwartet wird.

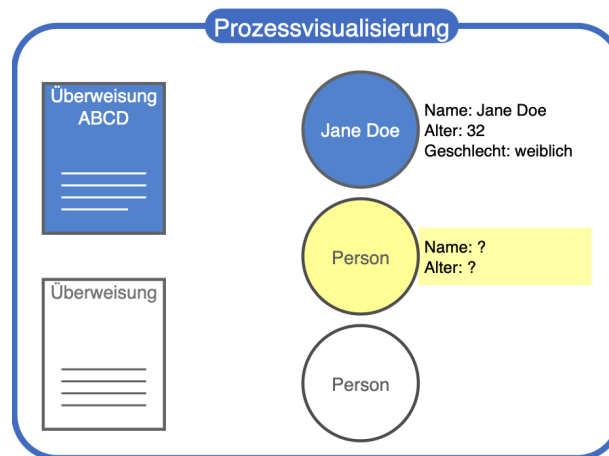


Abbildung 7.3: Einführung weiterer Prozesselement-Typen

Für ein Individuum vom Typ *Dataobject* und die damit via *constit_of* verknüpften *Attribute*-Individuen wird ein kombiniertes Element zur Darstellung eingeführt. Zur Erinnerung: Die Differenzierung zwischen *Dataobjects* und *Attributes* innerhalb der Wissensbasis dient nur dem Zweck, die Ausdrucksmächtigkeit zu erhöhen, damit über die Existenz von *Attributes* Schlussfolgerungen möglich sind. Dieses künstliche Konstrukt wird nicht in die Prozessvisualisierung übertragen, sodass ein *Dataobject* inklusive seiner verfügbaren *Attributes* als ein Prozesselement dargestellt wird. Das blaue runde Element *Jane Doe* steht für ein eben solches Individuum mit drei bekannten Attributen zum *Namen*, dem *Alter* und dem *Geschlecht*. Die nicht blauen runden Elemente stehen für noch unbekannte *Dataobjects*, also Individuen mit einer *Placeholder*-Zuordnung. Mit der gelben Hervorhebung führen wir zudem die Darstellung eines weiteren geschlussfolgerten Prozess-

zustandes analog zu *unexecutable* ein. Hiermit wird gekennzeichnet, wenn ein beliebiges Prozesselement, hier am Beispiel eines *Dataobjects* und einer Teilmenge seiner *Attributes* exemplarisch dargestellt, als *goal relevant* inferenziert werden.

Weitere Symbole beispielsweise für *Condition*-Elemente (Raute) oder aggregierte Elemente (Hexagon) werden bei ihrem ersten Vorkommen im weiteren Verlauf noch eingeführt. Neben den Symbolen für einzelne Prozesselemente kann auch die Vernetzung der Elemente im Rahmen einer PV dargestellt werden. Während dies für die Prozess-Designer:innen im Rahmen der Prozessmodellierung zentrale Informationen sind, ist die Vernetzung für die Anwender:innen innerhalb einer Prozessinstanz weniger relevant. Insbesondere da der ODD-BP Ansatz auf einen kontrollflussorientierten Ablauf verzichtet und somit dargestellte Kanten zwischen Prozesselementen kaum einen wertschöpfenden Beitrag innerhalb einer Prozessausführung leisten. Entsprechend wird bei der nachfolgend beschriebenen Systemumsetzung bei der Modellierung einer Prozessdefinition eine vollständige Prozessvisualisierung inklusive aller Verknüpfungen realisiert, während die PV einer Prozessinstanz vollständig auf die Beziehungen zwischen Prozesselementen verzichtet.

7.3 Generelle Methoden der Modifikation

Eine Veränderung an einer Prozessdarstellung kann auf verschiedene Arten durchgeführt werden. Zum einen kann die Graphik der Prozesselemente hinsichtlich Anordnung und Notation angepasst werden. Zum anderen kann die Darstellungsform und die Sicht der Prozesselemente modifiziert werden.

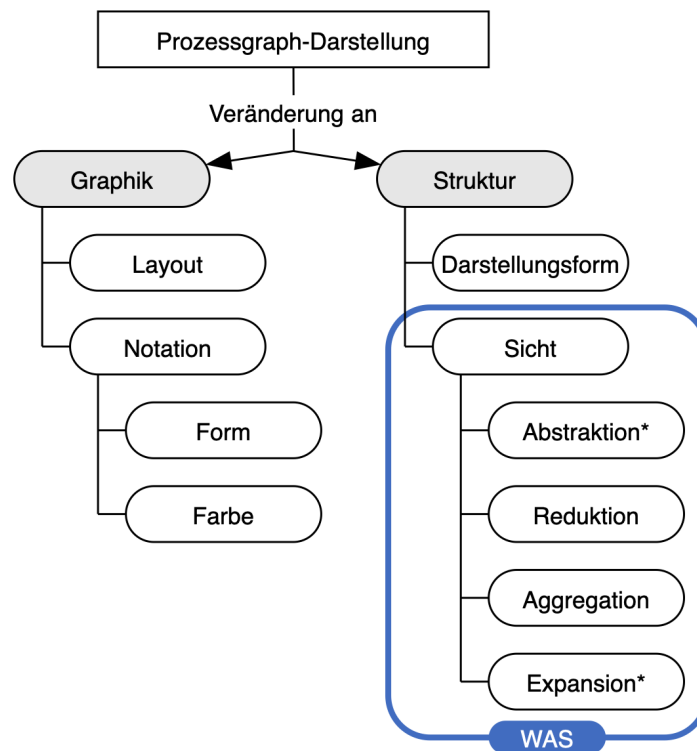


Abbildung 7.4: Freiheitsgrade der Gestaltung einer Prozessvisualisierung

Abbildung 7.4 zeigt eine Übersicht der Freiheitsgrade bei der Gestaltung einer Prozessvisualisierung nach Bobrik [20], erweitert um zwei zusätzliche Optionen (gekennzeichnet mit *).

Im Fokus dieser Arbeit steht dabei weniger die Frage, wie etwas dargestellt wird (Layout und Notation). Auch die Darstellungsform beschränkt sich auf die Variante eines Prozessgraphen und ignoriert zusätzliche oder alternative Darstellungsformen wie Swimlanes, Tabellen-, Matrix-, oder Gantt-Darstellungen. Stattdessen wird darauf fokussiert, was innerhalb eines Prozessgraphen inhaltlich dargestellt werden sollte. Hierzu werden nachfolgend vier grundlegende Modifikatoren eingeführt und definiert.

7.3.1 Abstraktion

Der erste Modifikator beeinflusst den Detailgrad der Darstellung eines beliebigen Prozesselementes. In einer sehr detaillierten Darstellung könnten zu einem Prozesselement alle bekannten Details dargestellt werden. Alternativ kann auch eine abstraktere Darstellung gewählt werden, bei der lediglich ein Merkmal angezeigt wird, das die Identität des Prozesselements repräsentiert. In einer noch abstrakteren Darstellung könnte das Element lediglich durch eine einfache Form mit einem gekürzten Bezeichner angezeigt werden.

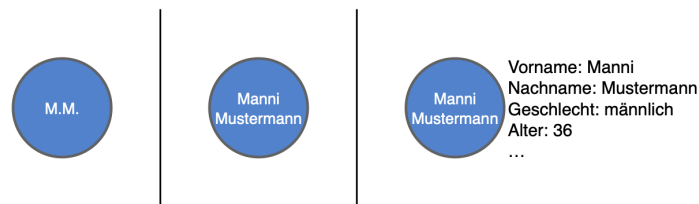


Abbildung 7.5: Darstellung eines Datenelements in drei Abstraktionsstufen

Abbildung 7.5 zeigt exemplarisch ein Datenelement in drei unterschiedlichen Abstraktionsstufen. Abhängig von der Relevanz des jeweiligen Prozesselementes zu einem gegebenen Zeitpunkt oder zu einer gegebenen Situation kann so der Abstraktionsgrad angepasst werden.

7.3.2 Reduktion

Eine weitere Möglichkeit, den Detailgrad einer Prozessdarstellung zu beeinflussen, ist das Entfernen einzelner Prozesselemente. Diese Form der Modifikation wird häufig verwendet, um beispielsweise Systemaktivitäten oder vom System erforderliche Zwischenschritte, welche keinen eigentlichen Beitrag zum Verständnis eines Prozessablaufs leisten, aus der Darstellung zu

entfernen. Es handelt sich dabei zumeist nicht um einen dynamischen Vorgang, wobei einzelne Elemente aufgrund einer individuellen Relevanz angezeigt oder entfernt werden, sondern häufig um eine generelle Filteroperation der Darstellung. Beispiele dieses Modifikators werden in unterschiedlichen Arbeiten [95, 62, 19], vor allem im Kontext kontrollflussorientierter Prozesse vorgestellt.

Im Rahmen dieser Arbeit soll der Modifikator jedoch nicht pauschal, sondern abhängig vom jeweiligen Kontext verwendet werden. Wird in einem spezifischen Prozesszustand und abhängig von dem aktuellen Betrachter festgestellt, dass ein Prozesselement eine Relevanz unter einem definierbaren Schwellwert hat, so wird das Element aus der Prozessdarstellung entfernt.

7.3.3 Aggregation

Eine weitere Möglichkeit, den Detailgrad einer Prozessdarstellung zu reduzieren, besteht darin, eine Gruppe von Prozesselementen zu einem einzigen repräsentativen Element zusammenzufassen. Im Gegensatz zu einem Reduktionsschritt, bei dem die Elemente komplett entfernt werden, ist das repräsentierende Symbol weiterhin sichtbar und die Anwender:innen können jederzeit damit interagieren, sodass bei Bedarf auch auf die aggregierten Elemente wieder zugegriffen werden kann. In bestehenden Prozessvisualisierungsansätzen wird der Aggregationsschritt zumeist verwendet, um Aktivitäten zusammenzufassen, die einem engen Weg innerhalb eines Kontrollflusses folgen und sich nicht verzweigen [20]. Aufgrund der ODD-BP zugrundeliegenden semantischen Prozessbeschreibung können darüber hinaus Elemente mit einer ähnlichen Bedeutung gruppiert und aggregiert werden. So können z.B. Daten-Elemente mit einem identischen Elementtyp durch ein einziges Element ersetzt werden, das die einzelnen Daten-Elemente in der Visualisierung substituiert.

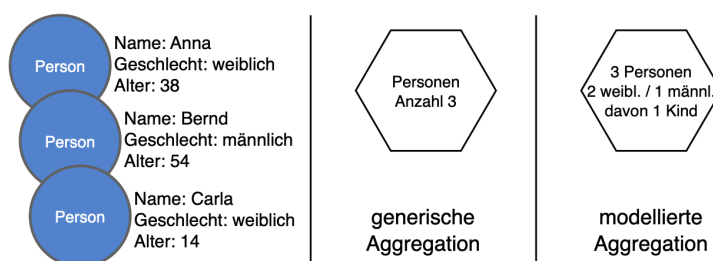


Abbildung 7.6: Darstellung zwei unterschiedlicher Aggregationsmöglichkeiten

Abbildung 7.6 zeigt auf der linken Seite drei Datenobjekte des gleichen Elementtyps „Person“. Jedes Datenobjekt hat einen Namen, das Geschlecht und das jeweilige Alter der Person. In der Mitte wird eine generische Aggregation dargestellt. Hierbei wird der Elementtyp „Person“, der gleichzeitig als Gruppierungsmerkmal verwendet wird, als neuer Bezeichner des aggregierten Elements (Hexagon) verwendet und zudem wird die Anzahl der darin zusammengefassten Elemente angegeben. Alle für diesen Aggregationsschritt erforderlichen Informationen sind innerhalb des semantischen Prozessmodells bekannt und können bei Bedarf verwendet werden. Anmerkung: Die Pluralform zu Person könnte über einen Thesaurus ermittelt werden.

Alternativ dazu ist, wie auf der rechten Seite abgebildet, auch eine Aggregation denkbar, welche aus den einzelnen Detailelementen wichtige Werte extrahiert und zu einer „höherwertigen“ Darstellung verarbeitet. Da für eine solche Aggregation weitere Kenntnisse über die inhaltlichen Zusammenhänge der zusammengefassten Elemente erforderlich sind, wird hierfür eine gezielte Modellierung des anzuwendenden Aggregationsschrittes erforderlich. An dieser Stelle wird die Möglichkeit einer modellierten Aggregation nicht weiter vertieft, sollte jedoch im Rahmen weiterführender Arbeiten untersucht werden.

7.3.4 Expansion

Die naheliegendste Möglichkeit, eine Darstellung anzupassen, ist die Herabsetzung der Granularität von Prozesssegmenten. Alternativ können aber auch zusätzliche Informationen in die Anzeige aufgenommen werden, welche nicht Bestandteil des ursprünglichen Prozessmodells sind. Der vierte Modifikator nutzt hierzu die Möglichkeiten der semantischen Prozessdefinition und der zugrundeliegenden Ontologien aus, um über definierte oder geschlussfolgerte Beziehungen den Detailgrad bei Bedarf gezielt zu erhöhen.

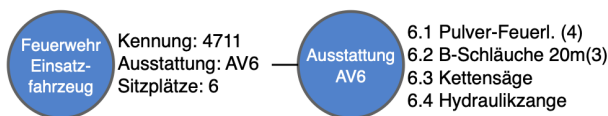


Abbildung 7.7: Darstellung einer Expansion

Abbildung 7.7 zeigt eine solche Expansion, indem zu dem Prozesselement „Feuerwehr Einsatzfahrzeug“ im Falle einer besonders hohen Relevanz des Elements ein weiteres bekanntes und in Beziehung stehendes Datenelement „Ausstattung AV6“ angezeigt wird. Da nicht jede Beziehung zu prozessexter-

nen Elementen zu einer Bereicherung der Prozessdarstellung für die Anwender:innen führt, sollten Beziehungen, die einen Mehrwert schaffen können, innerhalb der Ontologie gekennzeichnet werden. Insofern kann in Analogie zur modellierten Aggregation auch von einer modellierten Expansion gesprochen werden.

Mit Abstraktion, Reduktion und Aggregation stehen drei Modifikatoren zur Verfügung, um die Granularität von weniger relevanten Prozesssegmenten zu verringern und eine vereinfachte Visualisierung zu erreichen. Mit der Expansion steht zudem ein Modifikator zur Verfügung, um wichtigen Prozesselementen weitere Details hinzuzufügen. Mit diesen Methoden soll so die Aufmerksamkeit der Anwender:innen auf die derzeit wichtigen Prozesselemente gelenkt werden, mit dem Ziel diese bei der Ausführung wissensintensiver Prozesse zu unterstützen. Im Zentrum steht dabei die Herausforderung, die Relevanz jedes Prozesselements möglichst präzise einschätzen zu können, um die eingeführten Modifikatoren zielgerichtet einsetzen zu können. Hierzu wird nachfolgend ein entsprechendes Verfahren zur Adaption der Prozessdarstellung vorgestellt.

7.4 Adaptionverfahren

Das Adaptionverfahren kann in 4 Segmente gruppiert werden, welche sequentiell abgearbeitet werden und aufeinander aufbauen. Das Ergebnis ist eine adaptive Prozessvisualisierung, weshalb der Output des Verfahrens auch aPV abgekürzt wird.

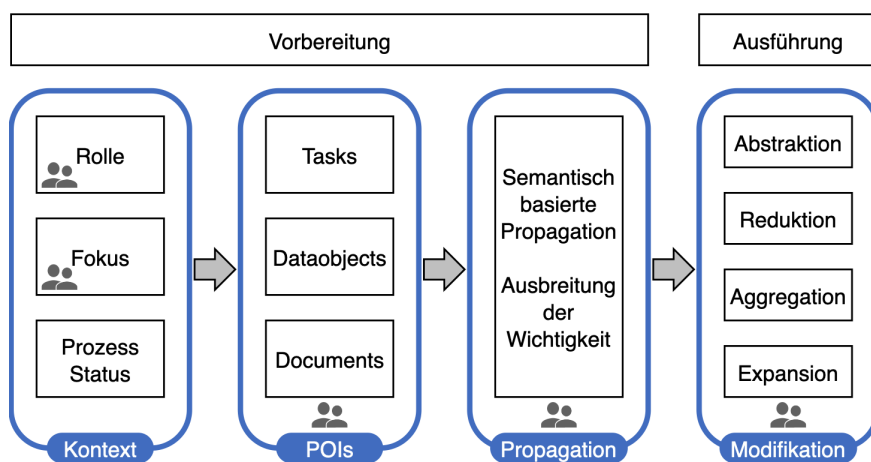


Abbildung 7.8: Verfahren einer adaptierten Prozessvisualisierung

Die ersten drei Segmente widmen sich der Vorbereitung, während im vierten Segment die Ausführung der Modifikationen erfolgt.

7.4.1 Kontext

Im ersten Segment geht es zunächst um die Berücksichtigung des derzeitigen Prozesskontextes. Anknüpfend an Kapitel 6 werden die durch Inferenzierung identifizierten Prozesszustände berücksichtigt. Hierbei definiert die Summe aller Prozesszustände der einzelnen Prozesselemente den generellen Status der jeweiligen Prozessinstanz. Zudem wird der Kontext durch die Rolle des Anwenders beeinflusst, für den die Prozessdarstellung bestimmt ist. So werden beispielsweise Aktivitäten, welche von den jeweiligen Anwender:innen durchgeführt werden können, bedeutender sein, als Aktivitäten, welche einer anderen Personengruppe zugeordnet sind. Darüber hinaus wird der Kontext durch den jeweiligen Fokus der nutzenden Person beeinflusst. Wurde bei-

spielsweise ein bestimmtes Element über das Benutzerinterface selektiert, so wird dieses Element als besonders relevant eingestuft. In Kombination aus der Rolle der jeweils bedienenden Person, dem aktuellen Fokus sowie dem generellen Prozesszustand ergibt sich der Kontext, auf Basis dessen die Adaption der Prozessdarstellung durchgeführt wird.

7.4.2 POIs

Im zweiten Segment werden die Ergebnisse aus dem ermittelten Kontext aufgegriffen und dazu verwendet die Elemente mit besonderem Interesse (engl. Points of Interest - POIs) zu identifizieren. Dies können Tasks sein, welche derzeit ausführbar sind und von der nutzenden Person ausgeführt werden können. Es kann sich dabei auch um Datenobjekte handeln, welche generell für Anwender:innen von besonderer Bedeutung sind. So werden Buchhalter:innen möglicherweise immer bestimmte Kennzahlen als wesentlich einstufen, während Mitarbeiter:innen mit einer anderen Rolle diese Datenobjekte als wenig relevant beurteilen. Dokumente könnten auf eine ähnliche Art und Weise durch benutzerspezifische Vorgaben als POIs eingestuft werden. Alternativ könnte auch ein *ungesehen* Status für Anwender:innen gepflegt werden, aufgrund dessen neue Dokumente, die noch unbekannt sind, automatisch als POI betrachtet werden. Bei der Festlegung von POIs werden demnach sowohl der aktuelle Prozesskontext als auch rollen- oder anwenderspezifische Parameter berücksichtigt. Im Kapitel 7.5 werden hierzu Erweiterungen der Domain-Ontologie eingeführt, welche das Modellieren dieser Aspekte ermöglichen.

7.4.3 Propagation

Die identifizierten POIs sind die Ausgangspunkte mit höchster Relevanz für eine Prozessvisualisierung. Diese Relevanz für die POIs sowie generell für alle Prozesselemente (PE) wird nachfolgend als Wichtigkeit zur Prozessvisualisierung (PVW) bezeichnet und in einem Zahlenwert zum Ausdruck gebracht. Im dritten Segment geht es nun um die Ausbreitung dieser Wichtigkeit, sodass letztlich jedes Prozesselement eine eigene PVW-Einstufung erhält. Die nachfolgenden Formeln definieren im ersten Schritt die generellen PVW-Grenzwerte für jedes beliebiges Prozesselement PE sowie den PVW-Startwerte der POI-Elemente.

$$PVW_{PE} \in \mathbb{R} \quad \wedge \quad PVW_{PE} \in [0, 100] \quad (\text{R } 18)$$

$$PVW_{POI} = 100 \quad (\text{R } 19)$$

Ausgangspunkt der PVW-Berechnung ist die Annahme, dass Elemente mit einer direkten Beziehung (Relation) zu einem POI für die nachfolgende Prozessausführung eine höhere Wichtigkeit besitzen als Elemente, welche weiter entfernt sind und somit nur durch eine Kette an Beziehungen mit einem POI verknüpft sind. Die Umsetzung erfolgt über einen Faktor zwischen 0 und 1 der, multipliziert mit dem PVW eines verknüpften Elements, zu einem Erhalt ($F = 1$) oder zu einer Abwertung ($F < 1$) eines PVWs für jedes Prozesselement führt. Der Faktor F ist dabei das Ergebnis einer Funktion $Factor$ von R und A , wobei (R) für jeden Relationstyp und (A) für jede(n) Anwender:in einen individuellen Wert annehmen kann. Hierzu wird nachfolgend in Kapitel 7.5 eigens eine Erweiterung der Domänen-Ontologie vorgestellt. Hierdurch kann die Relevanzberechnung umfassend auf die Bedürfnisse und Anforderungen einzelner Prozessteilnehmer ausgerichtet werden, indem Relationen zu wichtigen Prozesselementen einen hohen Faktor ($F \leq 1$) und zu unwichtigen Prozesselementen einen niedrigen Faktor ($F > 0$) erhalten. Für den Faktor sowie den PVW-Wert wird somit definiert:

$$F \in \mathbb{R} \quad \wedge \quad F \in [0, 1] \quad (\text{R } 20)$$

$$PVW_{PE_i} = Factor(R(PE_j, PE_i), A) * PVW_{PE_j} \quad (\text{R } 21)$$

Bei der Berechnung eines PVW-Wertes muss berücksichtigt werden, dass Prozesselemente mit einer Reihe anderer Prozesselemente über unterschiedliche Beziehungen verknüpft sein können. Daraus ergibt sich, dass ein Prozesselement viele unterschiedlich hohe PVW-Werte erhalten kann. Da der höchste dieser Werte ausschlaggebend sein soll, wie das Element in der nachfolgenden Prozessvisualisierung dargestellt werden soll, wird dies dem nachfolgend dargestellten Pseudocode entsprechend berechnet.


```

input  : A //Anwender:in
input  : aPOI //Array POIs
input  : aPE //Array Prozesselemente
output : aPVW //Array PVWs

1 new Array aPVW;
2 for each PE in aPE do
3   | aPVW.put(PE, 0);
4 new Array Queue;
5 for each Elem in aPOI do
6   | Queue.put(Elem);
7   | aPVW(Elem) = 100;
8 while not Queue.Empty() do
9   | CE = Queue.First();
10  | for each RE in RelatedElements(CE) do
11    | tempPVWRE = Factor(R(CE,RE),A) * aPVW(CE);
12    | if tempPVWRE > aPVW(RE) then
13      | aPVW(RE) = tempPVWRE;
14      | if Queue(RE) IS NULL then
15        | Queue.put(RE);
16    | Queue.Remove(CE);
17 return aPVW;

```

Algorithmus 1 : Relevanzberechnung aller Prozesselemente

Zunächst wird das Output-Array mit allen Prozesselementen und dem Startwert 0 initialisiert (Zeile 1,2,3). Anschließend wird eine Queue erstellt (4) und jedes übergebene POI-Element wird in diesen Stapel aufgenommen (6) und mit dem PVW-Wert 100 initialisiert (7). Anschließend wird in einer Schleife die Queue abgearbeitet (8). Hierzu wird das erste Element (CE) aus der Queue genommen (9). Dabei wird für alle verknüpften Prozesselemente (10) ein neuer temporärer PVW-Wert berechnet (11). Falls dieser Wert größer als der bisherige PVW-Wert des Elements ist (12), wird der höhere Wert übernommen (13) und das entsprechende Element, wenn nicht schon vorhanden (14), in den Arbeitsstapel aufgenommen (15). Sobald die Schleife über alle verknüpften Prozesselemente (10-15) abgearbeitet ist, wird das Element (CE) aus der Queue entfernt (16). Als Ergebnis werden schlussendlich alle Prozesselemente mit einem neu berechneten PVW-Wert zurückgeliefert (17).

Nach Abarbeitung dieses Propagation-Verfahrens besitzen alle Prozesselemente den für sie maximal möglichen PVW-Wert, womit gleichzeitig der Vorbereitungsabschnitt abgeschlossen ist und nun mit der Ausführung der Modifikationen begonnen werden kann.

7.4.4 Modifikation

Unter Ausnutzung der in Kapitel 7.3 vorgestellten Modifikatoren wird die vollständige Prozessdarstellung für einzelne Anwender:innen angepasst. Hierbei werden erneut anwenderspezifische Kriterien berücksichtigt, indem individuelle Schwellwerte zur Anwendung der einzelnen Verfahren wie Abstraktion, Reduktion, Aggregation und Expansion verwendet werden. Dies erlaubt es festzulegen, ab welchem PVW-Wert ein Element eher detailliert oder abstrakt dargestellt werden soll. Sobald dann für eine Gruppe ähnlicher Prozesselemente der PVW-Wert unter eine vordefinierte Schwelle fällt, können diese Elemente durch ein einzelnes aggregiertes Element repräsentiert werden. Fällt der PVW-Wert weiter ab und unterschreitet den Schwellwert zur Reduktion werden die entsprechenden Elemente vollständig aus der Prozessdarstellung entfernt. Ebenso wird mit der Expansion verfahren. Ist ein PVW-Wert oberhalb eines vordefinierten Levels, können verfügbare Detailinformationen zu Individuen außerhalb der Prozessinstanz in die Darstellung mit eingefügt werden. In Abbildung 7.8 wird dies durch die fünf Anwendersymbole in den einzelnen Verfahrensschritten explizit zum Ausdruck gebracht. Während in der bereits eingeführten Basis- und Domain-Ontologie Anwender:innen hinsichtlich ihrer Rolle bereits in einem Prozessmodell abgebildet werden können, so fehlt es bislang noch an der Modellierbarkeit der anwenderspezifischen Parameter einer aPV.

7.5 Erweiterung der Ontologie

Ausgangspunkt jeder anwenderspezifischen Parametrisierung der Prozessdarstellung ist das Konzept *Actor* aus der Base-Ontologie oder einer Spezialisierung davon. Abbildung 7.9 zeigt eine entsprechende Erweiterung der Ontologie um anwenderspezifische Parameter.

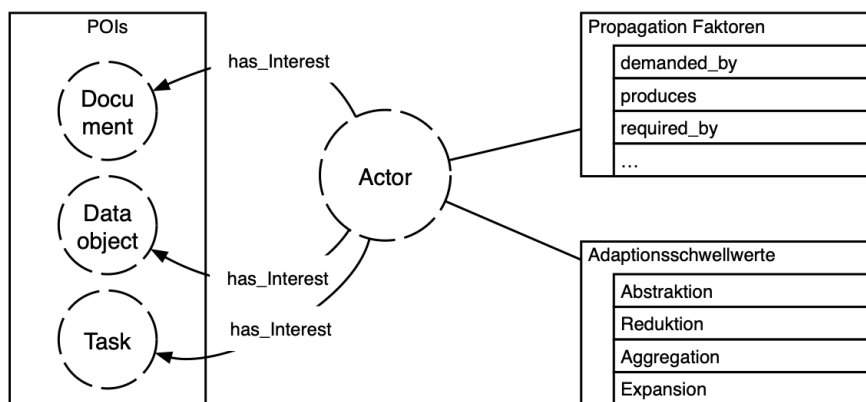


Abbildung 7.9: Erweiterung der Ontology um anwenderspezifische Parameter

Mit Blick auf den zweiten Vorbereitungsschritt, der Identifikation von POIs, kann durch die Relation *has_Interest* ein(e) Anwender:in ein besonderes Interesse an Individuen der Konzepte *Document*, *Dataobject* und *Task* zum Ausdruck bringen. Gleiches gilt entsprechend der Vererbungshierarchie natürlich auch für alle Spezialisierungen dieser Konzepte. So könnte modelliert werden, dass beispielsweise für Disponent:innen alle Individuen der Klasse *Verletzte Person*, welche eine Spezialisierung von *Person* ist, interessant sind. Mithilfe der Annotation *minImportance* zur Relation *has_Interest* kann nun festgelegt werden, welcher PVW-Wert einem entsprechend gekennzeichneten Prozesselement mindestens zugeschrieben werden soll. Falls unterschiedliche Intensitäten an Interesse modelliert werden sollen, können Spezialisierungen von *has_Interest* angelegt und mit unterschiedlichen *minImportance* Werten ausgestattet werden.

Im dritten Verarbeitungsschritt zur Verbreitung der Wichtigkeit zur Prozessvisualisierung wird entsprechend der eingeführten Formeln mit Faktoren zur Abwertung gearbeitet. Damit Anwender:innen ihre eigenen individuellen Faktoren und damit ihre individualisierte Berechnung der PVW Werte erhalten können, bedarf es einer entsprechenden Hinterlegung dieser Faktoren innerhalb der Ontologie. Die Umsetzung erfolgt, wie in Abbildung 7.9 dargestellt, durch ein Set an Annotationen zum Konzept *Actor* oder seiner Spezialisierungen, bei der zu jedem Beziehungstyp ein entsprechender Wert zwischen 0 und 1 hinterlegt werden kann. So kann beispielsweise auf die Input-Beziehungen zu Tasks (*required_by* und *demand_by*) durch einen hohen Faktor und damit einer geringen Abwertung besonderer Wert gelegt werden, während die Output-Beziehungen (*delivers* und *produces*) durch einen kleinen Faktor stark abgewertet werden könnten. Im Ergebnis werden so Elemente, die für einen Task benötigt werden, länger und detaillierter angezeigt, während das Ergebnis von Tasks früher ausgeblendet oder abstrakter abgebildet würde.

Im vierten Verarbeitungsschritt, in dem es zur Ausführung der vorgestellten Modifikatoren kommt, entscheiden Schwellwerte, ab welchem PVW-Wert eine jeweilige Anpassung der PV durchgeführt wird. Analog zu den Faktoren zur Propagation werden auch diese Schwellwerte als Annotation zum Konzept *Actor* oder seiner Spezialisierungen innerhalb der Ontologie hinterlegt. Die Schwellwerte liegen im gleichen Wertebereich zwischen 0 und 100 wie die PVW-Werte. Da die Modifikatoren eine unterschiedlich weitreichende Veränderung nach sich ziehen, ist eine entsprechende Abstufung der Schwellwerte nur naheliegend. So ist der Schwellwert für eine Abstraktion erwartungsgemäß höher als der Schwellwert für eine Aggregation oder Reduktion. Darüber hinaus müssen für manche Modifikatoren weitere Voraussetzungen erfüllt sein, damit diese eine Wirkung entfalten können. So kann eine Aggregation nur durchgeführt werden, wenn mehrere Elemente den entsprechenden Schwellwert unterschreiten, sodass diese sich für eine Aggregation aber nicht für eine Reduktion qualifizieren, während diese Elemente gleichzeitig dazu geeignet sein müssen, sich zu einem gemeinsamen substituierenden Element zusammenfassen zu lassen.

Wie dargelegt sind alle anwenderspezifischen Faktoren an das Konzept *Actor* oder eine Spezialisierung hiervon geknüpft. Damit nicht für jede Person eine umfangreiche Definition von Faktoren und Schwellwerten erforderlich wird, können solche Werte auf unterschiedlichen Ebenen innerhalb der Ontologie eingebunden werden.

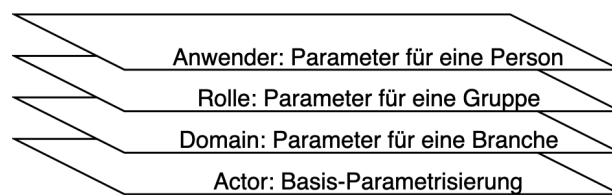


Abbildung 7.10: Schichtenmodell der anwenderspezifischen Parameter

Abbildung 7.10 zeigt ein Schichtenmodell, das eine allgemeingültige Basis-Parametrisierung für alle Anwender:innen erlaubt und gleichzeitig eine Individualisierbarkeit bis auf eine einzelne Person gestattet. Hierzu werden im ersten Schritt alle Parameter für das Hauptkonzept *Actor* eingelesen. Anschließend können aus der Domain-Ontologie über die Spezialisierung *Domain-Actor* Werte eingelesen werden, welche für eine Branche zu einer geeigneten Darstellung führen. Da Personen meist über eine oder mehrere Rollen in einem Prozess eingebunden sind, können in einer weiteren Spezialisierung *Role* die Werte für eine bestimmte Personengruppe gefunden werden. Letztlich kann als finale Spezialisierung eines *Actors* auch eine Person einzeln angelegt und mit individuellen Werten ausgestattet werden. Bei diesem Schichtenmodell gilt die Regel, dass jeder Wert, der unter einer Spezialisierung gefunden wird, den allgemeineren Wert der vorherigen Schichten überschreibt. Auf diese Weise können, aber nicht müssen, die Spezialisierungen eigene Werte aufweisen. Lediglich zum Konzept *Actor* sollten alle erforderlichen Werte als Basis-Parametrisierung hinterlegt sein.

7.6 Beispiel eines Adaptionvorgangs

Das eingeführte Adaptionsverfahren sowie die Methoden zu Modifikation werden nachfolgend an einem kompakten Beispiel zusammenfassend erläutert. Auf der linken Seite der Abbildung 7.11 wird eine vollständige PV mit einer Reihe an zusätzlichen Informationen dargestellt, sodass das zugrundeliegende Verfahren insbesondere mit Blick auf die durchgeführte Propagation nachvollzogen werden kann. Auf der rechten Seite wird das Ergebnis der Adaption, die aPV dargestellt.

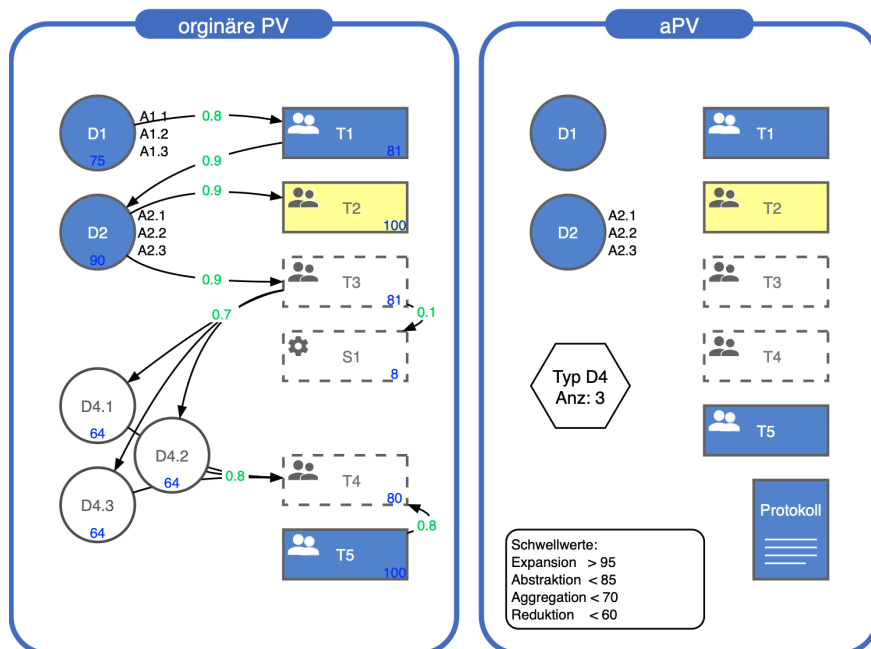


Abbildung 7.11: Adaption einer originären PV zu einer aPV

Zur Vorbereitung der Adaption ist wie definiert der Prozesskontext der initiale Ausgangspunkt. Aufgrund der Rolle einer Person und des aktuellen Prozesszustandes ist Task T2 von besonderer Bedeutung, da dieser unmittelbar ausgeführt werden könnte, dieser ausführbar ist und zudem als zielrelevant eingestuft ist. Des Weiteren wurde Task T5 selektiert, wodurch

der individuelle Fokus auf die PV zum Ausdruck gebracht wird. Ausgehend von diesen beiden kontextbezogenen Aspekten werden eben diese beiden Elemente als die aktuellen POIs festgelegt; entsprechend wird beiden ein PVW-Wert von 100 zugeordnet, ein Wert der bei allen Prozesselementen durch die blaue Zahl dargestellt wird. Dies definiert die beiden Startpunkte von denen ausgehend die Propagation stattfindet.

Zur besseren Lesbarkeit wird nicht im Einzelnen auf die Art der Beziehung zwischen den Elementen eingegangen. Stattdessen wird lediglich der anzuwendende Abwertungsfaktor, wie er aus den Annotationen der Ontologie ausgelesen werden kann, als Zahl in grüner Farbe unmittelbar auf den Kanten dargestellt. Zusätzlich wird zur Vereinfachung davon ausgegangen, dass dieser Abwertungsfaktor in beiden Richtungen identisch ist, also gleich ist für die zugrundeliegende Relation sowie sein inverser Konterpart.

Entsprechend der eingeführten Regeln wird so für D2 ein PVW-Wert von 90 ($100 \cdot 0,9 = 90$) ermittelt. Fortführend wird so auch ein jeweils individueller Wert für T1 und D1 berechnet. Gleiches gilt auch für den POI T5, der zur Berechnung der PVW-Werte von T4 und die drei Datenelemente D4.1, D4.2 und D4.3 maßgebend ist.

Interessant ist zudem T3, der als Beispiel dienen soll, wie zwei unterschiedliche PVW-Werte bei einem Element zusammenlaufen. Über die Verknüpfungskette zu T2 erhält T3 einen PVW-Wert von 81, während von T5 ausgehend ein PVW-Wert von rund 45 möglich wäre. Entsprechend den eingeführten Regeln setzt sich nur der höhere der beiden Werte durch.

Letztlich haben wir mit S1 noch einen Systemtask, der mit dem Faktor 0,1 eine sehr starke Abwertung erfährt, wodurch der sehr geringe PVW-Wert von 8 ermittelt wird.

Nachdem alle Prozesselemente ihren maximal möglichen PVW-Wert erhalten haben, kann die Adaption der PV durchgeführt werden. Hierzu sind die in der Ontologie hinterlegten Schwellwerte entscheidend, welche in Abbildung 7.11 als Legende eingefügt sind. Nachfolgend eine kurze Erläuterung zu jedem einzelnen Prozesselement.

- T2 wird aufgrund seines hohen PVW-Wertes unverändert dargestellt. Der Task würde sich grundsätzlich auch für eine Expansion qualifizieren, wenn ein geeigneter Kandidat zur Verfügung stünde.
- D2 hat mit $PVW=90$ kein Anspruch auf eine Expansion, wird aber in vollem Detailgrad weiterhin angezeigt.

- T1 erfüllt mit PVW=81 die Kriterien für eine Abstraktion der Darstellung, ein Aspekt der in diesem Beispiel für Tasks nicht unterstützt wird.
- D1 hingegen erfüllt mit PVW=75 ebenfalls die Kriterien für eine Abstraktion der Darstellung, was sich in dem Verzicht auf die Anzeige der Attribute A1.1, A1.2 und A1.3 widerspiegelt.
- T3 wird bei gleichem PVW-Wert wie T1 auch ebenso behandelt.
- S1 hat mit PVW-Wert=8 die Kriterien für eine Reduktion erfüllt und wird entsprechend vollständig aus der adaptierten Darstellung entfernt.
- D4.1, D4.2 und D4.3 haben mit einem PVW-Wert=64 nicht nur die Kriterien für eine Aggregation erfüllt. Zudem sind diese Elemente vom gleichen Typ und eignen sich somit für eine Substitution durch ein alternatives Element in Form eines Hexagons. Entsprechend dem Beispiel aus Abbildung 7.6 wird eine generische Aggregation dargestellt, weshalb das Hexagon lediglich den Typ der substituierten Elemente sowie die Anzahl angibt.
- T4 wird bei einem PVW-Wert von 80 ebenfalls wie T3 und T1 behandelt.
- T5 hingegen erfüllt mit einem PVW-Wert=100 wieder die Kriterien für eine Expansion und im Rahmen dieses Beispiels wird davon ausgegangen, dass außerhalb der Prozessinstanz ein Protokoll zu dem bereits ausgeführten Task T5 existiert, das in die PV aufgenommen und angezeigt wird.

Das Beispiel verdeutlicht nochmals die in diesem Kapitel eingeführte Methodik und das Verfahren zur Generierung einer adaptiven Prozessvisualisierung. Wie geschildert ist die Erwartungshaltung, dass die Anwender:innen bei der Prozessausführung unterstützt werden können, indem eine Darstellung mit einer erhöhten Aussagekraft erzeugt wird, indem unwichtigere Elemente entfernt oder mit geringerem Detailgrad angezeigt werden.

7.7 Fazit

Das grundlegende Ziel der vorgestellten Methodik einer aPV ist die Verbindung der zwei als gegenläufig beschriebenen Eigenschaften [84] von Flexibilität und Prozessunterstützung. Durch die Anpassung der Prozessvisualisierung werden den Anwender:innen vielversprechende oder bevorzugte Prozesselemente angeboten und damit letztlich als Vorauswahl von Handlungsoptionen empfohlen. Dabei bleibt grundsätzlich die Möglichkeit bestehen, jedwede Daten zu jedem beliebigen Zeitpunkt in eine Prozessinstanz einzubringen, während die aPV die Ausrichtung auf sich verändernde Prozessziele erlaubt und in Form von Empfehlungen eine substantielle Prozessunterstützung anbieten kann. Innerhalb von Kapitel 9 werden diese Aspekte im Rahmen einer empirischen Evaluation aufgegriffen.

8 ODD-BP System

Dieses Kapitel greift das methodische Grundgerüst der ersten drei Artefakte aus Kapitel 5, 6 und 7 auf und führt diese in Form einer Instanziierung des ODD-BP Ansatzes in einem vierten Artefakt zusammen. Diese prototypische Umsetzung des ODD-BP Systems wird nachfolgend in Kapitel 8.1 mit seiner zugrundeliegenden Systemarchitektur sowie den wesentlichen verwendeten Technologien vorgestellt. Damit sind alle grundlegenden Aspekte des ODD-BP Ansatzes in einem System realisiert, jedoch noch ohne einen spezifischen Domain- oder Anwendungsbezug herzustellen. Dies erfolgt im Kapitel 8.2, indem die für die Domäne Rettungsdienst, Feuerwehr und Notrufabwicklung zentralen Konzepte vorgestellt und innerhalb der Domain-Ontologie eingeführt werden. Anschließend wird in 8.3 der in Kapitel 4 vorgestellte Anwendungsfall aufgegriffen und als Referenz dienend als ein Basisprozess vorgestellt. Darauf aufbauend wird ein erweiterter Prozess gestaltet, der die mit ODD-BP angestrebten Vorteile nutzbar machen soll. Mit Blick auf die durchzuführende Evaluation wird zudem auch eine für das Anwendungsszenario geeignete adaptive Prozessvisualisierung vorgestellt. Anschließend wird der gesamte vorbereitete Systemaufbau in Kapitel 8.4 mit Hilfe eines Anwendungsbeispiels demonstriert.

8.1 Systemarchitektur

Für eine prototypische Umsetzung sind grundsätzlich eine Vielzahl an Realisierungsmöglichkeiten vorstellbar. Mit Blick auf den State-of-Art aktueller und verbreiteter Technologien und hinsichtlich einer zukünftigen Weiterführung des Forschungsthemas wurde eine betriebssystemunabhängige und zeitgemäße Client-Server Architektur auf Basis von Java / Java-Skript gewählt. Der Austausch zwischen Client und Server erfolgt über eine stateless REST-API, wodurch sichergestellt ist, dass jeder Zugriff autonom und ohne eine Session-Verwaltung abgearbeitet werden kann.

Mit der Trennung zwischen Client und Server geht eine Aufgabenteilung einher. Die grundlegende Aufgabe des Servers ist dabei die Umsetzung der

Funktionalitäten des ODD-BP Ansatzes, während dem Client die Aufgabe der Darstellung und der Interaktion der Anwender:innen mit dem System zukommt. Entsprechend dieser Aufgabenteilung ist auch der Datentransfer und die grundlegenden Verarbeitungsschritte innerhalb und zwischen den beiden zentralen Systembestandteilen konzipiert, welcher in Abbildung 8.1 dargestellt wird.

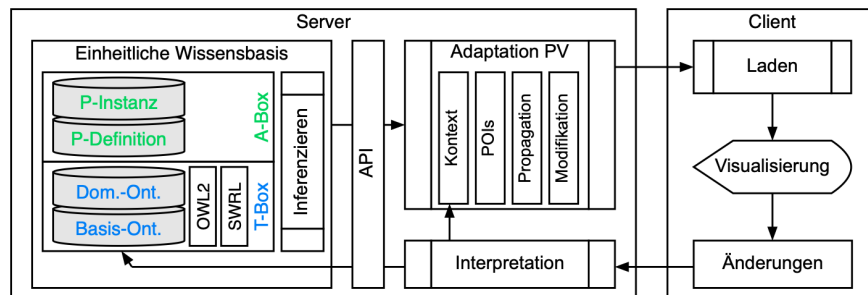


Abbildung 8.1: Datentransfer und Verarbeitungsschritte

Ausgangspunkt ist die einheitliche Wissensbasis, welche, wie in Kapitel 5.1 dargelegt, alle Informationsquellen, ausgehend von der Base-Ontologie und der Domain-Ontologie innerhalb der T-Box bis hin zu den Details zu Prozessdefinitionen und Prozessinstanzen innerhalb der A-Box, in einer einheitlichen Repräsentation zusammenfasst. Die grundlegende Methodik des ODD-BP Ansatzes wird durch die in Kapitel 5 und 6 eingeführten OWL2- und SWRL-Regeln als Teil der Basis-Ontologie realisiert. Darüber hinaus wird Expertenwissen, welches nachfolgend noch eingeführt wird, als zusätzliche SWRL-Regeln innerhalb der Domänen-Ontologie hinterlegt. Erweitert wird das explizit modellierte Wissen um implizites Wissen, welches mit Hilfe eines Inferenzierers geschlussfolgert wird. Auf diese Wissensbasis kann innerhalb des Servers über eine definierte API lesend und schreibend zugegriffen werden. Darüber hinaus sind gezielte Abfragen, z.B. via SPARQL möglich.

Ausgelesen werden alle zu einem einzelnen Prozess zugeordneten (*contains*) Prozesselemente, wobei es sich sowohl um Prozessdefinitionen als auch um Prozessinstanzen handeln kann. Als Ergebnis erhält man einen Prozessgraphen, der alle Prozesselemente als Individuen einschließlich aller Beziehungen zu den grundlegenden Konzepten bzw. anderen Individuen umfasst. Falls es sich bei dem Prozess um eine Prozessdefinition handelt, wird

der Prozessgraph in eine Datenstruktur zur Prozessvisualisierung überführt, welche im JSON Format abgerufen und übertragen werden kann, und wird vollständig zur Darstellung an den Client übergeben. Im Falle einer Prozessinstanz durchläuft die Datenstruktur der PV vor der Übergabe an den Client den in Kapitel 7 vorgestellten Verarbeitungsprozess zur Adaption der PV zu einer aPV.

Der Client erhält die Daten im JSON Format und überführt diese in eine zur Visualisierung geeignete Repräsentation. Die Visualisierung selbst kann in jedem HTML5 konformen Browser mit aktiver Java-Skript Unterstützung dargestellt werden. Jede Interaktion mit der Prozessdarstellung führt zu einer Übertragung von neuen Daten an den Server, welche ebenfalls in das JSON Format überführt werden. Dies kann einfache Daten zur Anzeige der Zoom-Stufe oder des Darstellungsbereich, einfache prozessbezogene Daten wie das aktuell selektierte Element bis hin zu weitreichenden Veränderungen der Prozessdaten umfassen. Je nach Art der übertragenen Daten führt dies zumindest zu einer Anpassung der Wissensbasis, ggf. aber auch zusätzlich zu einer Neueinschätzung des Prozesszustandes durch den Inferenzierer und in Folge dessen auch zu einer erneuten Verarbeitung des Prozessgraphen einschließlich einer aktualisierten Prozessvisualisierung, welche wieder an den Client übertragen wird.

8.1.1 Verwendete Technologien

Bei der Implementierung des Servers wurde auf eine Reihe etablierter und frei verfügbarer Technologien wie Software-Development-Kits (SDKs) und Komponenten mit vordefinierter API zurückgegriffen. Als Programmiersprache wurde JAVA verwendet und innerhalb der Entwicklungsumgebung Eclipse gearbeitet. Zur Versionsverwaltung und zur verbesserten Zusammenarbeit mit studentischen Hilfskräften wurde GitHub verwendet. Ein Apache Tomcat Server wurde sowohl für den Testbetrieb als auch auf den Entwicklungsrechnern zu Debug-Zwecken eingesetzt.

Zur Modellierung der Basis- und Domain-Ontologien sowie zur Definition und Validierung der Inferenzierungsregeln wurde mit Protégé 5.2.0 gearbeitet. Zum Inferenzieren selbst wurde mit dem Pellet-Reasoner gearbeitet, da dies im Rahmen durchgeführter Testreihen der einzige Open-Source Reasoner zu sein scheint, der das Schlussfolgern von mehrstufigen Beziehungsketten unterstützt. Diese Fähigkeit ist mit Blick auf die Schlussfolgerung des Status *goal-relevant* zwingend erforderlich und wurde von alternativen Reasonern wie Fact, Racer und HermiT nicht ermöglicht.

Die Daten zur Konzeptionalisierung (T-Box) sowie zur Instanziierung (A-Box) werden in drei Dateien im OWL-Format (bpbase.owl, bpdomain.owl, bpinstance.owl) persistiert. Diese Dateien werden über das Apache Jena Framework eingelesen und in ein integriertes Modell überführt, auf das über eine Java-API zugegriffen wird. Gleichzeitig kann innerhalb des Apache Jena Frameworks dieselbe Installation des Pellet-Reasoners, welche auch in Protégé verwendet wird, eingebunden und zum Inferenzieren eingesetzt werden. Die geschlussfolgerten Erkenntnisse werden Teil des Gesamtmodells und können so gemeinsam mit dem expliziten Wissen als eine zentrale Wissensbasis fungieren. Das Deployment des jeweiligen Entwicklungsstandes erfolgt über war-Dateien, welche an eine Apache Tomcat Installation übergeben, dort entpackt und als Webservice ausgeführt werden.

Der Client basiert grundlegend auf Angular als ein TypeScript-basiertes Front-End-Webapplikationsframework. Entsprechend basiert die Entwicklung der Anwendungsfunktionalität auf der Programmiersprache TypeScript sowie HTML- und CSS-basierten Deklarationen hinsichtlich der Darstellung. Damit folgt das Angular-Framework und somit auch der Client dem Model View Controller (MVC) Design Pattern. Hierbei wird die Anwendungslogik von dem Benutzerinterface isoliert, was der Übersichtlichkeit und Wartbarkeit der Webanwendung dient. Zur Darstellung wird auf das D3 Framework ¹ zurückgegriffen. Im Rahmen dieses Frameworks wird die Komponente „Force-Directed Graph“ angeboten, welche neben einer Graphdarstellung von Knoten und Kanten auch eine dynamische Anordnung der Elemente unterstützt. Mithilfe von Redux als eine Open-Source JavaScript-Bibliothek werden alle aktuellen Zustandsinformationen verwaltet sowie das Verhalten der Webanwendung bei einem Zustandswechsel realisiert. Für eine agile und nicht durch API-Zugriffe blockierte Webanwendung wird eine event-basierte Verarbeitung der Server-Kommunikationsaufrufe benötigt. Hierzu wird auf der RxJS Bibliothek aufgebaut und mit „Observables“ gearbeitet, was eine asynchrone Abarbeitung von Ereignislisten erlaubt.

Neben den genannten Kerntechnologien für Client und Server werden noch mehrere Komponenten mit einem weniger zentralen Anwendungsbezug (beispielsweise das dynamische Nachladen von Menüeinträgen oder ein Kontextmenü) verwendet, welche an dieser Stelle keine gesonderte Betrachtung verdienen.

¹<https://d3js.org>

8.1.2 Basisfunktionalität

Mit Blick auf die grundsätzliche Architektur und den für die Umsetzung eingesetzten Technologien sind innerhalb von Client und Server eine Reihe zentraler Basisfunktionalitäten realisiert worden, welche nachfolgend kurz vorgestellt werden.

Das entwickelte System unterstützt grundsätzlich die Darstellung von Prozessdefinitionen und Prozessinstanzen. Hierbei unterscheidet sich der Funktionsumfang und die Darstellung mit Bezug auf die unterschiedliche Aufgabenstellung, die mit den beiden Prozesstypen einhergeht.

Prozessdefinition

Bei einer Prozessdefinition wird der vollständige Prozessgraph mit allen Prozesselement, einschließlich aller Verknüpfungen untereinander zur Visualisierung vom Server an den Client übertragen. Der Client bietet dann die Option an, alle Verknüpfungen anzuzeigen oder alle Verknüpfungen auszublenden und nur bei einem Mouse-Over-Event oder der Auswahl eines einzelnen Elements dessen Verknüpfungen anzuzeigen. Über einen Auswahldialog können zudem neue Elemente in die Prozessdefinition eingefügt werden, wobei neben dem Elementtyp ein Elementname frei vergeben werden kann. Der Elementtyp basiert auf einem der Basis-Elementtypen (*Task*, *Dataobject*, *Document*) oder einer darauf aufbauenden Spezialisierung, welche in der Base-Ontologie oder der Domain-Ontologie definiert ist. Mit dem Einfügen wird innerhalb der A-Box ein Individuum der entsprechenden Klasse angelegt und dieses auch unmittelbar mit dem Individuum der aktuellen Prozessdefinition über einer *contains*-Beziehung verknüpft.

Mit dem Drücken der ALT-Taste wird dem Prozess-Designer die Möglichkeit geboten, mit der Maus zwei Prozesselemente miteinander zu verbinden. Hierbei entscheidet die Richtung (Start-Element, Ziel-Element) sowie der Typ der beiden Elemente, welchen Relationstyp die Verknüpfung annehmen kann. Stehen mehr als ein Relationstyp zur Verfügung, wird eine Auswahl angeboten, bei nur einem verfügbaren Relationstyp wird dieser sofort gewählt. Wird beispielsweise eine Verbindung von einem *Dataobject* (oder einer Spezialisierung davon) zu einem *Task* (oder einer Spezialisierung davon) gezogen, so kann der Relationstyp nur zu einer *required_by* Beziehung führen. Bei einer Verbindung von *Dataobject* zu *Dataobject* hingegen muss der Designer zwischen einer *has_a* oder einer *composed_by* Beziehung wählen. Diese generische Umsetzung des Designvorgangs hat den Vorteil, dass

mit einer Erweiterung oder Anpassung der Basis- oder Domain-Ontologie gleichzeitig das Werkzeug zum Modellieren einer Prozessdefinition seine Fähigkeiten aktualisiert.

Über ein Kontextmenü auf einem Prozesselement können die Bezeichner und Beschreibungen nachträglich editiert oder ein Element wieder gelöscht werden. Zudem kann ein Positionierungsmodus gewählt werden, der festlegt, wie ein Prozesselement in der Prozessdefinition aber auch später bei einer Prozessinstanz positioniert wird. Zur Auswahl stehen drei Modi: Standardmäßig werden Prozesselemente mit festen Koordinaten fixiert, sodass diese nur an einer eindeutigen Position angezeigt werden können. Alternativ kann die Position als ein flexibler Anker definiert werden, sodass sich Elemente in einem gewissen Bereich um den Anker selbst ausrichten können. Letztlich kann ein Element auch ohne Referenzpunkt und damit frei positioniert werden. Gerade beim zweiten und dritten Modus kann auf die Fähigkeiten des im Client integrierten Force-Directed Graph zurückgegriffen werden, wodurch die Elemente selbstständig versuchen, eine sich nicht überlappende Positionierung unter Berücksichtigung der Verknüpfungen, die wie Gummibänder fungieren, zu finden. Im Folgenden wird auf diese Funktionalität nicht weiter eingegangen, allerdings kann sich die Fähigkeit einer dynamischen Positionierung der Elemente im Kontext sehr dynamischer Prozesse und Prozessdarstellungen in anderen Anwendungsszenarien als wertvoll erweisen.

Hervorzuheben ist noch die Modellierung von Beziehungen zwischen *Attributes* und *Tasks*. Wie in Kapitel 5.2.1 dargelegt, spielt zur Inferenzierung der Prozessstatus auch die Verfügbarkeit einzelner Attribute eine besondere Rolle. Zur Modellierung wird hierzu dem Prozess-Designer die Möglichkeit geboten, auf jeder *required_by* sowie jeder *delivers* Beziehung zwischen *Dataobjects* und *Tasks* zusätzliche *Attributes* zu definieren, welche für den *Tasks* erforderlich sind oder von diesem geliefert werden. Zudem werden zwei grundsätzlich unterschiedliche Arten von *Attributes* unterstützt. *Attributes* ohne Vorgabe erlauben eine freie Eingabe von Werten (z.B. Name einer Person), während *Attributes* mit einer Vorgabe nur die Auswahl vordefinierter Werte (z.B. Geschlecht: männlich, weiblich, divers, unbekannt) unterstützen.

Abbildung 8.2 zeigt exemplarisch einen Ausschnitt einer Prozessdefinition. Entsprechend der in Kapitel 7 eingeführten Symbolik sind auf der linken Seite fünf spezialisierte *Task*-Elemente abgebildet. Diese *Tasks* dienen der Datenakquise und liefern entsprechend ein Ergebnis an die datentragenden Elemente in Form von Kreisen. Für eine bessere Übersicht sind alle Bezie-

lungen ausgeblendet bis auf eine. Diese *delivers* Beziehung von einem *Task* zum *Dataobject* ist ausgewählt, woraufhin auf der rechten Seite weitere Details dieser Verknüpfung definiert werden können. So ist dort ergänzend definiert, dass der Task ein bestimmtes *Attribute*, das Alter, liefern soll.

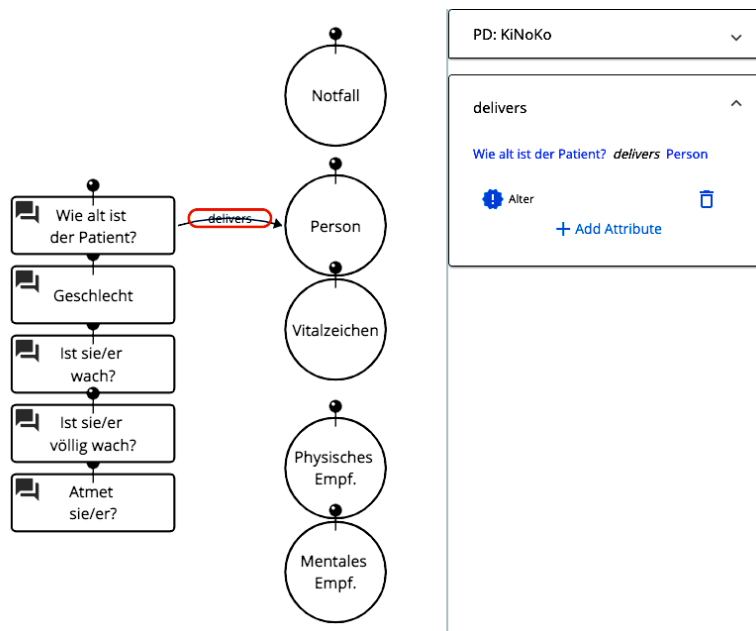


Abbildung 8.2: Ausschnitt einer Prozessdefinition

Die Abbildung greift einige der zuvor beschriebenen Aspekte der Basisfunktionalität hinsichtlich Prozessdefinitionen auf und vermittelt einen Eindruck, wie ein Prozess-Designer seine Modellierungsaufgabe, unterstützt durch eine übersichtliche Prozessvisualisierung, erfüllen kann.

Prozessinstanz

Eine Prozessinstanz ist eine Instanziierung einer vorbereiteten Prozessdefinition. Entsprechend wird zum Zeitpunkt der Erstellung eine Kopie aller Individuen einschließlich aller entsprechenden Verknüpfungen und Klassenzuordnungen erzeugt. Hinzu kommt, dass entsprechend der in Kapitel 5 getroffenen Regeln alle Prozesselemente, mit Ausnahme der Elemente vom Typ *Condition*, der Klasse *Placeholder* zugewiesen werden.

Mit Blick auf die Funktionalitäten zur Modellierung einer Prozessdefinition sind die Funktionen für die Ausführung einer Prozessinstanz völlig unterschiedlich. So werden zunächst keine Veränderungen an dem Prozessmodell unterstützt, weshalb es auch keine Optionen gibt, neue Elemente hinzuzufügen, Verknüpfungen zu verändern oder Löschoptionen durchzuführen. Stattdessen wird der Anwender in die Lage versetzt, Tasks auszuführen und Daten einzugeben oder zu editieren. Jede Veränderung wird wie beschrieben vom Client in ein Datenpaket im JSON-Format an den Server gesendet und wird dort über die API von Apache Jena in der Wissensbasis persistiert. In der Regel triggert eine solche Veränderung auch den Inferenzierungsmechanismus, woraufhin der Prozessstatus aller Prozesselemente neu bestimmt wird. Der Zyklus folgt dann dem in Abbildung 8.1 abgebildeten Datenaustausch zwischen Client und Server.

Der Verarbeitungsschritt des Inferenzierens umfasst dabei neben dem offensichtlichen, dem Aktivieren des Reasoners, auch die Ausführung der in Kapitel 6.4 eingeführten Agenten zur Zustandsbestimmung der *Condition* Elemente. Hierbei werden die Ergebnisse der Agenten in Form von Statements als explizites Wissen zurück in die A-Box geschrieben. Da dies die Grundlage bestehender Schlussfolgerungen verändern kann, erfordert jede Anpassung durch einen Agenten auch das erneute Aktivieren des Reasoners. Dieser Ablauf birgt grundsätzlich das Risiko in einer endlosen Verarbeitungskette fest zu hängen, weshalb bei der Umsetzung eine maximale Anzahl dieser Schleifendurchläufe festgelegt wurde.

Mit Blick auf die angestrebte Evaluation der Methodik durch das prototypische System ist zudem die Dauer dieses Verarbeitungsschrittes von Bedeutung. Hierbei muss festgehalten werden, dass die Zeiten des Pellet-Reasoners extrem abhängig sind von der Anzahl der Instanzen innerhalb der A-Box. Bei wenigen Individuen (20) präsentiert sich das System mit einer Bearbeitungszeit von unter einer halben Sekunde noch halbwegs agil (ein einzelner Durchlauf des Pellet-Reasoners - ohne eine durch Agenten getriggerte Schleife). Aber bereits bei 100 Individuen schwankte je nach Konfiguration die Bearbeitungszeit zwischen 3-10 Sekunden. Zudem sind die Zeiten abhängig von der Anzahl der Instanzen innerhalb der A-Box, nicht der Anzahl von Prozesselementen des jeweils aktuellen Prozesses.

Aufgrund dieser langen Bearbeitungszeiten ist parallel ein zweiter Java-basierter Inferenzierungsmechanismus (Fast-Inf) implementiert worden, welcher hardcodiert die festgelegten Regeln aus den Ontologien umsetzt und alternativ zum Pellet-Reasoner aktiviert werden kann. Die Möglichkeit

des Umschaltens zwischen den beiden Reasonern erlaubt es, die korrekte Arbeitsweise des Fast-Inf-Reasoners jederzeit zu überprüfen. Bei aktiviertem Pellet-Reasoner werden dessen Ergebnisse verwendet, während gleichzeitig die Ergebnisse von Fast-Inf erzeugt und mit den bestehenden Pellet-Ergebnissen verglichen werden. Die Verarbeitungszeiten der Fast-Inf Variante liegen konstant bei unter 200 Millisekunden, auch bei komplexeren Prozessinstanzen mit deutlich über 100 Prozesselementen. Zudem werden die Verarbeitungszeiten nicht durch andere Individuen innerhalb der A-Box aber außerhalb der aktuellen Prozessinstanz beeinflusst. Bei einer mehrfachen Aktivierung aufgrund von Veränderungen durch Agenten liegt der gesamte Verarbeitungsschritt des Inferenzierens dennoch meist deutlich unter einer Sekunde, was eine Grundvoraussetzung für die spätere Evaluation erfüllt.

Nach dem Inferenzierungsschritt werden die in Kapitel 7 eingeführten Methoden zur anwenderbezogenen Adaption der Prozessvisualisierung angewendet. Hierbei wird der Prozessgraph einer Prozessinstanz modifiziert, bevor dieser zur Anzeige an den Client weitergeleitet wird. Hierzu wird insbesondere im Zusammenhang mit aggregierten Prozesselementen eine weitere Funktionalität erforderlich. Da aggregierte Elemente keine Repräsentanz innerhalb der Wissensbasis besitzen, existiert auch kein Kennzeichner über den ein solches Element in einer Interaktion mit Anwender:innen identifiziert werden kann. Zu diesem Zweck wird auf Basis der Kennzeichner der aggregierten Elemente eine temporäre ID für das neue Prozesselement erzeugt. Auf diesem Weg können Anwender:innen ein entsprechend neu geschaffenes aggregiertes Element anklicken und der Server kann diese Interaktion identifizieren und bei der nächsten Prozessvisualisierung durch eine veränderte Festlegung der POIs berücksichtigen.

Auch hier wurden nur die wesentlichen Funktionalitäten des umgesetzten Systems zusammenfassend vorgestellt, um einen Eindruck zu der grundsätzlichen Arbeitsweise dieser Client-Server Lösung zu vermitteln. Alle beschriebenen Technologien und Funktionalitäten sind dabei domainunabhängig und können in beliebigen Anwendungsszenarien eingesetzt werden.

8.2 Modellierung des Domänenwissens für die Anwendung

Zur Modellierung eines Prozesses im eingeführten Anwendungsszenario der Notrufkommunikation bedarf es zunächst einiger grundlegender domainspezifischer Konzepte, welche nachfolgend eingeführt werden. Diese wurden in Zusammenarbeit mit dem ärztlichen Leiter der ILtS-Ludwigshafen spezifiziert und werden entsprechend der bisherigen Sprachregelung ebenfalls mit englischen Bezeichnungen benannt. Darüber hinaus wird die Domain-Ontologie um zusätzliche Konzepte erweitert, welche Anwender:innen mit einer gewissen Form von künstlicher Intelligenz bei der Durchführung einer Notrufabfrage unterstützen soll. Zu diesen Konzepten gehören auch Regeln, welche nutzstiftendes Expertenwissen repräsentieren und exemplarisch vorgestellt werden.

8.2.1 Domainspezifische Konzepte

Im Rahmen einer Notrufkommunikation können fünf Arten von Prozess-elementen identifiziert werden: Fragen (*QuestionTasks*), personenbezogene Daten (*Person* und *PersonConditions*), umgebungsbezogene Daten (*Environment*) und Handlungen (*ActionTasks*).

QuestionTasks als eine Spezialisierung von *UserTask* sind der generelle Ausgangspunkt einer Notrufabfrage. Der Prozess-Designer kann beliebig viele Individuen dieser Klasse in der Prozessdefinition anlegen, um jeweils unterschiedliche Fragen nach Alter, Geschlecht und Zustand einer Person zu modellieren. Die Ergebnisse dieser Fragen führen zu Daten, welche unter Instanzen verschiedener *Dataobject*-Typen persistiert werden. Genau genommen werden diese Daten in den Attributen der entsprechenden Klasse abgespeichert. So sind einer *Person* (eine direkte Unterklasse von *Dataobject*) über die *consists_of* Beziehung *Attributs* wie *Name*, *Age*, *Gender* zugeordnet. Für eine verkürzte Schreibweise wird nachfolgend zu einem Datenobjekt mit einem Doppelpunkt getrennt ein zugehöriges Attribut dargestellt (*Dataobject:Attribute* / *Person:Gender*). Neben den personenbezogenen Daten wird der Zustand der jeweiligen Person über Spezialisierungen zu *PersonConditions* (PC) entsprechend der nachfolgenden Liste adressiert.

- Vital-Zustand
PC_vital:consciousness, :breath, :breath_movement, ...

- Physisches Empfinden
PC_physical:pain, :fever, :fade_skin, :cold_sweat, ...
- Mentales Empfinden
PC_mental:weakness, :stress, :threaten, :fear, ...
- Internistisch
PC_internal:infection, :hearth, :diabetes, :cholesterol, :hypertension, ...
- Chirurgisch
PC_surgical:fracture, :combustion, :bleeding, ...
- Neurologisch
PC_neurological:headache, :dizziness, :nervous_system, ...

Mittels *ActionTasks* als eine weitere Spezialisierung von *UserTask* können für Anwender:innen vordefinierte Dispositionsarten festgelegt werden. Der Prozess-Designer kann so eine Reihe von Individuen dieser Klasse innerhalb der Prozessdefinition anlegen und damit die Disposition eines einfachen Krankentransports, den Einsatz eines Rettungstransports, einen Notarzteinsatz oder einen Rettungshubschrauber zur Verfügung stellen.

8.2.2 KI-Support Konzepte und Regeln

Mit den eingeführten Konzepten kann bereits die standardisierte oder strukturierte Notrufabfrage (SNA) als Prozess modelliert werden. Zur Erweiterung und zur Unterstützung der Anwender:innen durch ontologisches Expertenwissen werden jedoch noch weitere Konzepte eingeführt. Abgestimmt auf den in der Evaluation durchgeführten experimentellen Ablauf, in dem drei unterschiedliche Notfälle simuliert werden, werden entsprechend auch drei neue Konzepte eingeführt, welche gemeinsam als KI-basierte Support-Konzepte (KIS-Konzepte) oder im Falle von Instanziierungen als KI-basierte Support-Elemente (KIS-Elemente) bezeichnet werden. Bei den KIS-Konzepten handelt es sich um eine Spezialisierung des Typs *Dataobject*, zu denen über die zugeordneten *Attributes* gezielt weitere Erkenntnisse und Einschätzungen abgebildet werden sollen. Mit Blick auf den experimentellen Aufbau (siehe Kapitel 9.2) werden folgende drei KIS-Konzepte eingeführt.

- *SeverityLevelFall*: Mit diesem Konzept wird das Ziel verfolgt, den Schweregrad eines Sturzes einschätzen zu können.

- *UncertainCauseUnconsciousness*: Dieses Konzept dient der Suche nach der Ursache einer Bewusstlosigkeit.
- *SuspicionHearthAttach*: Mittels diesem Konzept wird angestrebt, den Verdacht eines Herzinfarktes erhärten oder ausräumen zu können.

Nachfolgend wird die grundsätzliche Methodik hinter den KIS-Konzepten erläutert. Eine detaillierte umgangssprachliche Beschreibung der Regeln für jeden der drei KIS-Typen erfolgt im Kapitel Evaluation unter 9.2.

Jedes KIS-Konzept erhält innerhalb der Domain-Ontologie SWRL-Regeln zur Aktivierung, Verbreitung und Einschätzung des jeweils repräsentierten Sachverhalts. Exemplarisch werden diese SWRL-Regeln am Beispiel des Verdacht eines Herzinfarktes (*SuspicionHearthAttach* - SHA) erläutert.

Aktivierung

Die Aktivierung eines SHA-Elementes erfolgt, indem ein entsprechendes Element vom Typ *SuspicionHearthAttach* als *ProcessGoal* eingestuft wird, sobald innerhalb des Prozesses definierte Attribute mit spezifischen Werten identifiziert werden können. Für eine verkürzte Schreibweise und eine bessere Lesbarkeit wird bei der Darstellung der SWRL-Regeln auf das notwendige Prefix (Domain:) verzichtet. Entsprechend der eingeführten Konzepte ist bereits dokumentiert, wann das Prefix *bpbase:* für die Base-Ontologie und wann das Prefix *bpdomain:* für die Domain-Ontologie erforderlich wäre. Formel (R 22) definiert die SWRL-Regel zum Aktivieren eines SHA-Elements und für eine bessere Verständlichkeit wird nachfolgend jede Zeile nochmals in ihrer Bedeutung beschrieben.

$$\begin{aligned}
& Process(?p) \wedge SHA(?sha) \wedge contains(?p, ?sha) \wedge \\
& SHA_assessment(?shaRes) \wedge consistsOf(?sha, ?shaRes) \wedge \\
& PC_physical(?phys) \wedge contains(?p, ?phys) \wedge \\
& PCP_painLocation(?pain) \wedge consistsOf(?phys, ?pain) \wedge \\
& valueStr(?pain, ?val) \wedge swrlb : matches(?val, "Brust") \\
& \rightarrow ProcessGoal(?sha) \wedge ProcessGoal(?shaRes)
\end{aligned} \tag{R 22}$$

1. Zeile: Es existiert ein *Process*(?p) und ein KIS-Element vom Typ *SHA*(?sha) und (?sha) gehört in den *Process*(?p).

2. Zeile: Es existiert ein *Attribute*(?shaRes), das zu *Dataobject*(?sha) gehört.
3. Zeile: Es existiert ein *Dataobject* PC_physical(?phys), das zum *Process*(?p) gehört.
4. Zeile: Das *Dataobject*(?phys) besitzt ein *Attribute* vom Typ PainLocation(?pain).
5. Zeile: Dieses *Attribute*(?pain) hat den Wert „Brust“ erhalten.
6. Zeile: Wenn alle Bedingungen erfüllt sind, dann definiere (?sha) und (?shaRes) als *ProcessGoal*.

Entsprechend der zuvor eingeführten Regeln wird das KIS-Element (?sha) und das zugehörige *Attribute* (?shaRes) auch *goal-relevant*, sofern die Elemente noch der Gruppe *Placeholder* zugeordnet sind.

Verbreitung

Auf Grund der bekannten iterativen Ausbreitung von *goal-relevant* durch den Prozessgraphen können nun alle Prozesselemente ebenfalls *goal-relevant* werden, sofern diese zur Erreichung der Elemente (?sha) und (?shaRes) etwas beitragen können. Hierzu ist es lediglich erforderlich, dass innerhalb der Prozessdefinition eine entsprechende Verknüpfungen zwischen (?sha) und relevanten anderen *Tasks* und *Dataobjects* vom Prozess-Designer modelliert worden sind. Alternativ kann eine Verbreitung auch durch weitere Regeln innerhalb der Domain-Ontologie festgelegt werden. Formel R 23 zeigt eine solche Regel, welche festlegt, dass im aktuellen Fall eines Verdachts auf einen Herzinfarkt die Information bezüglich einem bereits bekannten hohen Blutdruck wichtig ist.

$$\begin{aligned}
& Process(?p) \wedge SHA(?sha) \wedge contains(?p, ?sha) \wedge \\
& SHA_assessment(?shaRes) \wedge consistsOf(?sha, ?shaRes) \wedge \\
& goal - relevant(?shaRes) \wedge \\
& PC_internal(?int) \wedge contains(?p, ?int) \wedge \\
& PCI_hyperthension(?hyp) \wedge consistsOf(?int, ?hyp) \\
& \rightarrow ProcessGoal(?hyp)
\end{aligned} \tag{R 23}$$

Auf diese Weise kann eine Ausbreitung sowohl durch die modellierte Prozessdefinition als auch durch allgemeingültige Regeln innerhalb der Domain-Ontologie festgelegt werden. Auch hier gilt: Aus einem *ProcessGoal* wird nur

der Status *goal-relevant*, wenn das entsprechende Prozesselement noch als *Placeholder* eingestuft ist. Ist das entsprechende Element bereits bekannt, wird die Relevanz nicht verändert.

Einschätzung

Auf dem gleichen Wege wie die Werte einzelner Attribute zur Aktivierung verwendet werden, so können Attribute-Werte auch für eine systembasierte Einschätzung eines Sachverhalts verwendet werden. Mit der nachfolgenden Formel R 24 wird ausgewertet, ob wir zum vorliegenden Fall Kenntnis über eine hohen Blutdruck haben. Ist dies der Fall, wird das entsprechende KIS-Element mit einem eigens hierfür geschaffenen Label *Tag_Likely* markiert. Hierdurch wird gekennzeichnet, dass der Verdacht auf einen Herzinfarkt berechtigt ist und entsprechende Maßnahmen ergriffen werden sollten.

$$\begin{aligned}
& Process(?p) \wedge SHA(?sha) \wedge contains(?p, ?sha) \wedge \\
& SHA_assessment(?shaRes) \wedge consistsOf(?sha, ?shaRes) \wedge \\
& goal - relevant(?shaRes) \wedge \\
& PC_internal(?int) \wedge contains(?p, ?int) \wedge \qquad (R\ 24) \\
& PCI_hyperthension(?hyp) \wedge consistsOf(?int, ?hyp) \wedge \\
& valueStr(?hyp, ?val) \wedge swrlb : matches(?val, "hoch") \\
& \rightarrow tag_likely(?sha)
\end{aligned}$$

Über das zugeordnete Label können nun aus der Domain-Ontologie Symbole oder Texte aus den Annotationen ausgelesen und zum KIS-Element angezeigt werden, um Anwender:innen weitere Hilfestellungen anzubieten. Bei der Systemeinschätzung handelt es sich jedoch lediglich um einen Vorschlag. Anwender:innen können nun dieser Systemeinschätzung folgen oder wenn nötig von dieser Einschätzung abweichen und eine andere eigene Entscheidung explizit festlegen. Diese Entscheidung ist dann maßgebend und kann vom Prozess-Designer unmittelbar mit Dispositionsempfehlungen verknüpft werden.

Die vorgestellte Methodik im Zusammenhang mit den KIS-Elementen und ihrer Aktivierung, Verbreitung und Einschätzung lässt erkennen, wie das innerhalb der Domain-Ontologie integrierte Expertenwissen im Zuge einer Prozessausführung in Aktion treten und eine gezielte Unterstützung für Anwender:innen bieten soll.

8.3 Prozessmodelle des Anwendungsszenarios

Mit den eingeführten Konzepten ist der Rahmen geschaffen, innerhalb dessen nun die Modellierung des zu evaluierenden Anwendungsfalls erfolgt. Grundsätzlich werden hierbei zwei Prozessmodelle angestrebt. Ein Prozessmodell, um das etablierte Verfahren der SNA mit den für die Evaluation vorbereiteten Notfällen testen und den Ablauf der Notrufe protokollieren zu können, nachfolgend als Basis-SNA (bSNA) bezeichnet. Ein weiteres Prozessmodell, welches die durch ODD-BP neu geschaffenen Möglichkeiten in einem erweiterten Verfahren ausnutzen kann, nachfolgend als erweiterte SNA (eSNA) bezeichnet. Dies dient dem grundsätzlichen Ziel, den etablierten bSNA-Prozess mit einem optimierten eSNA-Prozess vergleichen und die angestrebten Verbesserungen validieren zu können.

Um unerwünschte Einflüsse bei diesem Vergleich möglichst ausschließen zu können, wird bewusst nicht auf das etablierte System, die Cobra-Leitstellen-Lösung der Firma ISE, zurückgegriffen. Stattdessen wird sowohl das etablierte Verfahren der bSNA als auch das erweiterte Verfahren der eSNA innerhalb des prototypisch entwickelten ODD-BP Systems umgesetzt. Darüber hinaus wurden zudem beide Verfahren in ein einziges Prozessmodell integriert, um keine unerwünschten Modellierungsabweichungen zwischen zwei getrennten Prozessdefinitionen zu riskieren. Im Rahmen der Ausführung einer Prozessinstanz wird dann gezielt die erweiterte Funktionalität der eSNA aktiviert oder deaktiviert.

Die bSNA entspricht dabei dem in Kapitel 4 Abbildung 4.1 dargestellten Ablauf. Eine Abweichung von der darin fest vorgegebenen Reihenfolge wird natürlich auch innerhalb der bSNA Umsetzung innerhalb des ODD-BP Systems nicht unterstützt. Lediglich die eSNA Umsetzung erlaubt Abweichungen von dem vorgegeben Pfad und erweitert die Notrufabfrage um die zuvor eingeführten KIS-Elemente.

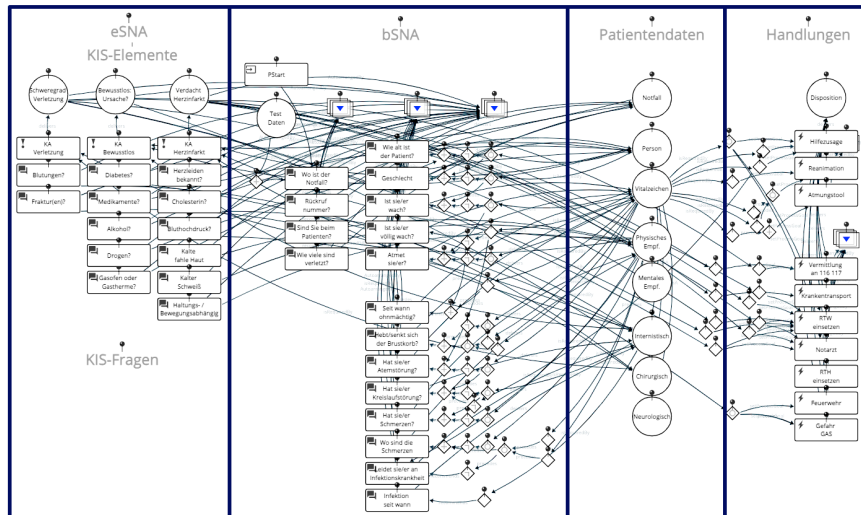


Abbildung 8.3: Vollständiges Prozessmodell der bSNA und eSNA

Abbildung 8.3 bietet einen Überblick über das vollständige und kombinierte Prozessmodell und soll lediglich einen Eindruck von dem generellen Aufbau bieten. In der Darstellung sind einzelne Bereiche gekennzeichnet, innerhalb derer die zum Bezeichner zugehörigen Prozesselemente gruppiert sind.

Die bSNA einschließlich der Patientendaten und Handlungen besteht aus insgesamt 162 Prozesselementen, den 18 Fragen (Rechtecke mit Sprechblasen) in der Mitte, 10 Datenobjekten (Kreise) zu den Patientendaten, 10 Handlungsoptionen (Rechtecke mit Blitzsymbol) sowie 61 Condition-Elementen (Rauten) zur Steuerung des vordefinierten Ablaufs. Nicht explizit in der Prozessvisualisierung dargestellt sind die 63 Attribute, welche auf die 10 Datenobjekte verteilt sind und über die Verknüpfungen mit den *Tasks* und *Condition*-Elementen in den Prozessablauf eingebunden sind.

Die eSNA erweitert den Prozess zunächst lediglich durch die drei KIS-Datenelemente (Kreise). Da im Kontext dieser drei Prozesselemente Daten benötigt werden, welche nur teilweise durch den Fragenkatalog der bSNA adressiert werden, sind noch weitere 16 KIS-Fragen modelliert worden.

Die dargestellten Verknüpfungen zwischen den Prozesselementen zeigt die ausgeprägte Interaktion, vor allem zwischen Individuen der Klasse *Task* und *Dataobject*. Die vielen Condition Elemente sind vor allem für die Umsetzung der bSNA erforderlich, um durch gezielte Constraints innerhalb des

deklarativen Prozessmodells einen zum Kontrollfluss identischen Ablauf umsetzen zu können. Bei einer reinen Umsetzung unter dem durch ODD-BP geprägten Prinzips und mit etwas mehr Freiheit und Eigenverantwortung des Disponenten könnten alle Elemente im Abschnitt bSNA durch wenige (5-6) gezielte Frage-Elemente, welche gleich mehrere Attribute liefern können, ersetzt und somit das Prozessmodell wesentlich vereinfacht werden. Die Komplexität ist somit wesentlich auf das Remodellieren des bSNA Ablaufs zurückzuführen.

Nachfolgend werden die einzelnen Bereiche nochmals im Detail dargestellt und hinsichtlich ihrer Aufgabe beschrieben.

Abbildung 8.4 zeigt die den Disponent:innen zur Verfügung stehenden Dispositions- und Handlungsoptionen an. Hierbei muss hervorgehoben werden, dass diese Handlungen jederzeit ausführbar sind, unabhängig von dem jeweiligen Prozesszustand der bSNA oder eSNA. Die ersten drei Optionen (Hilfezusage, Reanimation, Atmungstool) sind Werkzeuge für die Disponent:innen selbst und starten bei Bedarf zusätzliche Module und Dialoge. Anschließend folgen direkte Zuordnungen von Hilfskräften, angefangen mit der Vermittlung zur ärztlichen Hotline 116 117, über die Aktivierung von Fahrzeugen und Personal (Krankentransport, RTW-Rettungstransportwagen, Notarzt, RTH-Rettungstransporthubschrauber, Feuerwehr). Abschließend können noch gezielte Handlungsempfehlungen an den Anrufer weitergegeben werden, hierzu exemplarisch ein Hinweis bei einer Gefahr durch Gas-Austritt.

Die *Condition*-Elemente (Rauten) sind Teil der eSNA und führen zu gezielten Handlungsempfehlungen durch das System, sobald die Daten eine vordefinierte Situation repräsentieren. Alternativ könnte auch dies aus dem Prozessmodell entfernt und als allgemeingültige Regeln innerhalb der Domain-Ontologie definiert werden.

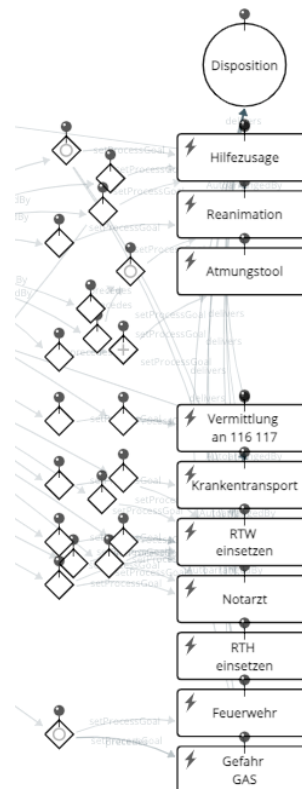


Abbildung 8.4: Handlungen

Abbildung 8.5 gibt einen Überblick über die zentralen Dataobjects der Prozessdefinition. Dabei werden über das erste Element *Notfall* generelle Informationen zum Notruf erfasst wie der Ort des Notfalls, die Rufnummer des Anrufers, ob dieser selbst betroffen ist oder z.B. vor Ort dabei ist. Mit dem *Dataobject* Person werden allgemeine aber relevante Daten erfasst wie der Name, das Geschlecht und das Alter der Person oder ob die Person eine besondere Situation hat wie eine Schwangerschaft oder beispielsweise eine Abhängigkeit von einem Rollstuhl. Hinter dem *Dataobject* Vitalzeichen werden die grundlegenden Vitaldaten zu einer Person hinterlegt. Dieses Element ist in der Abbildung selektiert (roter Rahmen), woraufhin dem

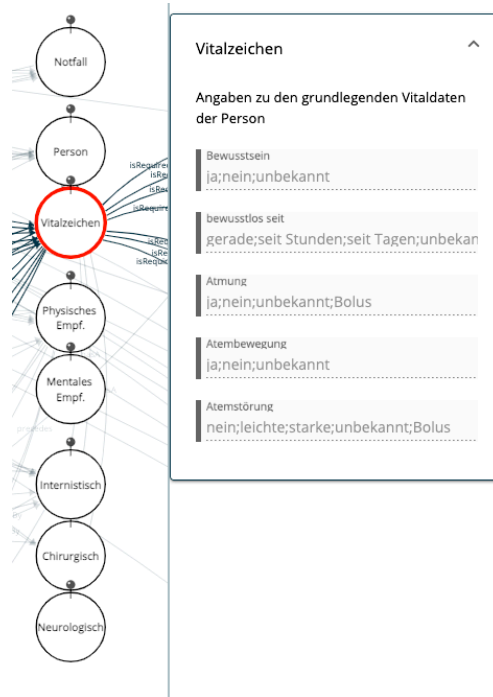


Abbildung 8.5: Daten

Prozess-Designer die zugehörigen Attribute rechts angezeigt werden. Diese Attribute werden befüllt durch einzelne Frage-Elemente oder sie werden von *Condition*-Elementen zur Steuerung des Prozessablaufs verwendet. Im Falle von Attributen mit vordefinierten Werten werden dem Prozess-Designer diese Werte in Form einer mit Semikolon getrennten Auflistung angezeigt. So wurden innerhalb der Domain-Ontologie für das Attribut Bewusstsein die Werte „ja“, „nein“ und „unbekannt“ vorgegeben. Die weiteren *Dataobject* Elemente entsprechen den zuvor in Kapitel 8.2 eingeführten domainspezifischen *PersonConditions* Konzepten.

Die dargestellten Prozesselemente (*Dataobjects* und *Attributes*) stellen in diesem wissenszentrierten und datenorientierten Prozess die zentralsten und wesentlichsten Elemente dar. An ihrem Zustand orientiert sich die gesamte Prozessausführung, was durch die umfangreichen Verknüpfungen zu anderen Prozesselementen untermauert wird.

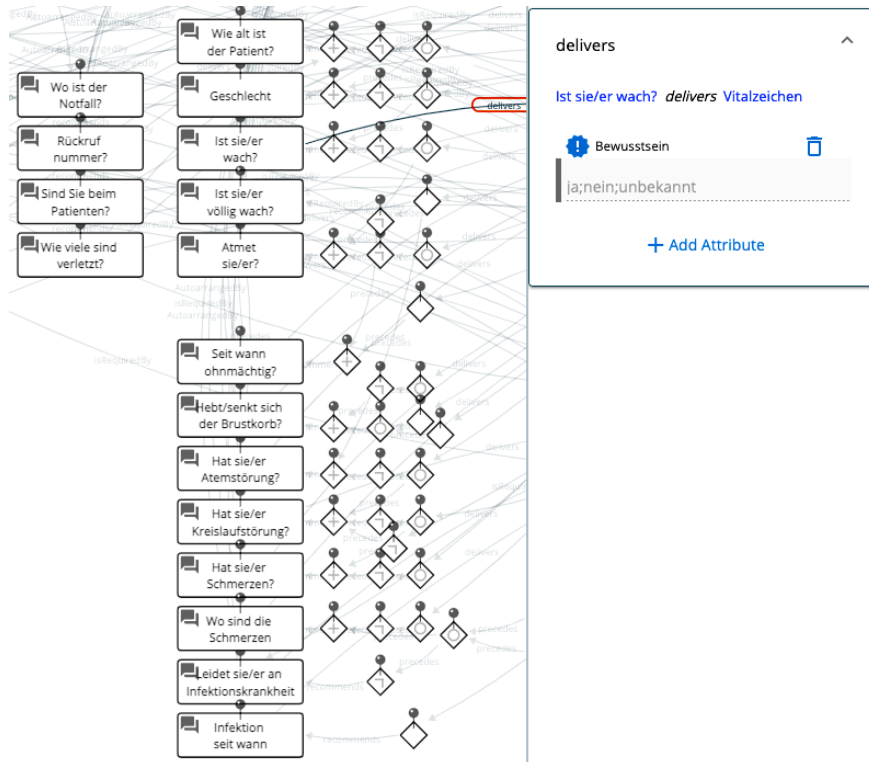


Abbildung 8.6: bSNA

Mit Abbildung 8.6 wird der Ausschnitt der bSNA-Fragen nochmals im Detail abgebildet. Im Vergleich mit dem in Kapitel 4 vorgestellten Anwendungsfall wird erkennbar, dass die dort aufgeführten Fragen sich hier in einer vergleichbaren Reihenfolge wiederfinden. Zunächst kommen allgemeine Fragen zum Ort des Notfalls, der Rückrufnummer der anrufenden Person und zum generellen Umstand des Notrufs und anschließend Fragen zum Notfall selbst. Diese Fragen liefern als Ergebnis die Daten und Attribute der auf der vorherigen Seite eingeführten Prozesselemente.

Exemplarisch wurde die *delivers*-Beziehung vom Task „Ist sie/er wach?“ selektiert, woraufhin der Prozess-Designer auf der rechten Seite die Option erhält, die Beziehung zu dem verknüpften *Dataobject* „Vitalzeichen“ genauer zu spezifizieren und wie dargestellt das Attribut „Bewusstsein“ ebenfalls über eine *delivers*-Beziehung zu verknüpfen. Hier könnten nun auch weitere

Attribute zu dem gleichen *Dataobject* verknüpft werden oder eine solche Verknüpfung auch wieder gelöscht werden.

Abbildung 8.7 komplettiert als viertes Segment der vorgestellten Prozessdefinition den Bereich der eSNA. Im Zentrum stehen dabei die drei KIS-Elemente „Schweregrad Verletzung“, „Bewusstlos: Ursache?“ und „Verdacht Herzinfarkt“. Darüber hinaus sind zu den für diese KIS-Elemente zusätzlich erforderlichen Attribute weitere KIS-Fragen (Rechtecke mit Sprechblasen) definiert worden. Zur jeweiligen Aktivierung dieser Attribute und Fragen, also der Einstufung als *goal-relevant*, wurde für das Anwendungsszenario keine Umsetzung über die Domain-Ontologie gewählt, sondern direkt über die Prozessdefinition reali-

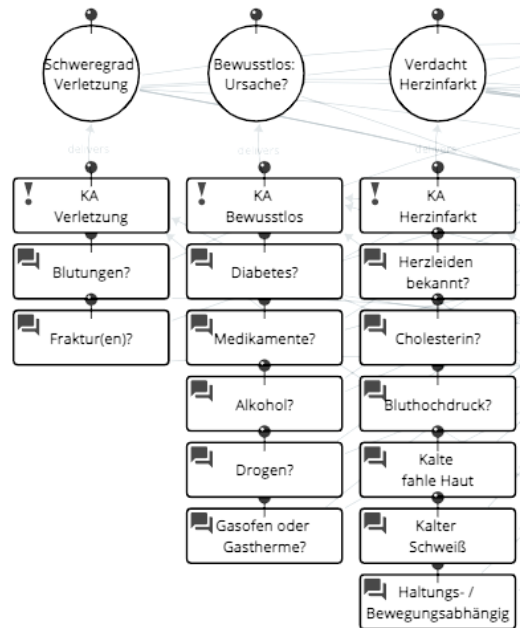


Abbildung 8.7: eSNA

siert. Dies wird durch die Prozesselemente „KA Verletzung“, „KA Bewusstlos“ und „KA Herzinfarkt“ erreicht. Das Kürzel KA steht dabei für die Basisklasse *Knowledge-Akquisition*, einer Spezialisierung vom Typ *Systemtask*. Über die *delivers*-Beziehung zum KIS-Element wird die darin angestrebte Einschätzung (ein Attribut) referenziert. Solange diese fehlt, überträgt sich die *goal-relevance* auf das KA-Element und von dort auf alle als relevant betrachteten Attribute unterschiedlicher Datenobjekte. Im nächsten Schritt wird dann diese *goal-relevance* auch auf die KIS-Fragen übertragen, wodurch sich alle dargestellten Prozesselemente danach richten, ob die angestrebte Einschätzung innerhalb des KIS-Elementes erfolgt ist oder noch fehlt.

Während bislang nur die Prozessdefinition des Anwendungsszenarios und damit die Sicht des Prozess-Designers dargestellt wurde, wird nachfolgend die Umsetzung der bSNA und eSNA als Prozessinstanz aus der Sicht der Disponent:innen in der Leitstelle betrachtet.

8.4 System Demonstration

Der Ablauf einer Notrufabfrage unter Ausnutzung des definierten Prozessmodells wird nachfolgend exemplarisch an einem fiktiven Notruf vorgestellt. Zum Starten wird hierzu eine neue Prozessinstanz als Kopie der eingeführten Prozessdefinition erzeugt und wie beschrieben werden alle Prozesselemente, außer den *Condition*-Elementen, auch dem Konzept *Placeholder* zugeordnet, womit ihre Funktion als Platzhalter für die erwarteten oder angestrebten *Tasks*, *Dataobject* und *Attributes* repräsentiert wird. Nachfolgend wird dann dieser Prozess in drei Zuständen (unmittelbar nach dem Prozessstart, bei Erfüllung der bSNA, unmittelbar vor Prozessende nach eSNA) dargestellt und der Prozesszustand, die Prozessdarstellung sowie der Bezug zum zugrundeliegenden Anwendungsszenario erläutert.

8.4.1 Prozessstart

Der Prozess startet mit der Frage nach dem Ort des Notfalls. Mit Eingabe der Antwort wird dann zur nächsten Frage gewechselt, der Frage nach der Rückrufnummer, unter der die anrufende Person nach Abbruch der Verbindung wieder erreicht werden könnte.

Abbildung 8.8 repräsentiert diesen Prozesszustand und zeigt, wie dies gegenüber den Disponent:innen nach Durchführung der adaptiven anwenderbezogenen Prozessvisualisierung (aPV) dargestellt wird. Analog zur Anordnung innerhalb der Prozessdefinition kann auch hier in drei Bereiche unterschieden werden. Auf der linken Seite finden sich alle Fragen, welche sich der Aufnahme der, den Notfall betreffenden, Situation widmen. In der Mitte werden alle erfassten Daten zusammengetragen und zu jedem *Dataobject* werden die bedeutungstragenden Attribute angezeigt. Auf der rechten Seite haben die Disponent:innen jederzeit die Möglichkeit, eine Disposition auszuwählen. In einem abgesetzten Bereich auf der rechten Seite, der aufgrund der besseren Lesbarkeit in einem separaten grauen Rahmen in die Darstellung eingefügt wurde, kann das jeweils selektierte Prozesselement bearbeitet werden. Hierzu wird neben dem Bezeichner des Prozesselements

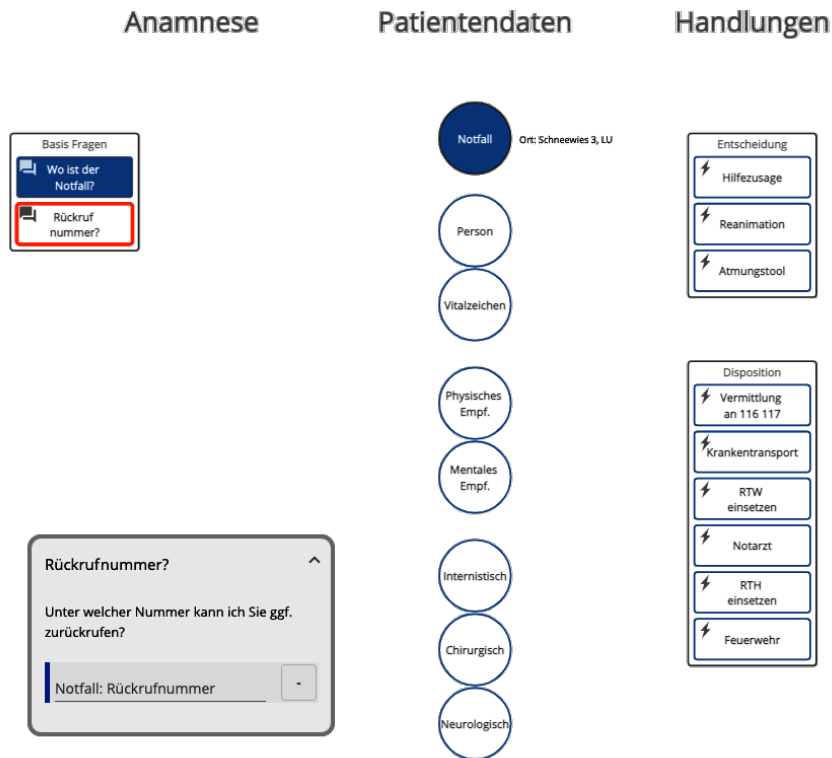


Abbildung 8.8: aPV einer Prozessinstanz nach Eingabe der ersten Frage

auch eine Beschreibung angezeigt. Darunter erfolgt die Eingabe von Daten oder die Auswahl vordefinierter Optionen.

Zur Vereinfachung der Anzeige wird auf die Darstellung der Kanten zwischen den Prozesselementen verzichtet. Darüber hinaus wird umfassend von der Möglichkeit der Reduktion Gebrauch gemacht, wodurch im dargestellten Beispiel nur noch eine einzige ausführbare Frage „Rückrufnummer?“ angezeigt wird. Zur Verdeutlichung des Fortschritts werden alle bereits ausgeführten Tasks weiterhin angezeigt und entsprechend der eingeführten Regeln hinsichtlich Art und Status der Prozesselemente visualisiert.

Die Regeln zur Visualisierung sind Teil der Domain-Ontologie und in Form von Annotationen zu den einzelnen Prozesselement-Typen hinterlegt.

So wird gesteuert, dass die Patientendaten immer angezeigt werden, unabhängig von ihrem Zustand. Gleiches gilt für die Handlungsoptionen auf der rechten Seite. Im Gegensatz dazu ist für die Fragen definiert, dass ein gewisser Status (*executable* oder *executed*) erforderlich ist, um mit dem PVW-Wert über der Schwelle für eine Reduktion zu liegen.

Die strikte Reduktion auf die Anzeige einer einzelnen Frage folgt dem Vorbild der bSNA-Umsetzung innerhalb des Cobra-Leitstellensystems. Auch dort wird jeweils immer nur eine Frage dargestellt und nach deren Beantwortung wird dem Entscheidungsbaum folgend die jeweils nächste Frage angezeigt. Insofern folgt der Ablauf dem gleichen Schema, wodurch sich hinsichtlich der angestrebten Evaluation eine Vergleichbarkeit der Prozessergebnisse untersuchen lässt, ohne dass eine Beeinflussung durch die Verwendung unterschiedlicher Systeme riskiert wird.

8.4.2 Prozessinstanz nach bSNA und vor eSNA

Die exemplarische Prozessinstanz hat den vorgesehenen Ablauf nach bSNA durchlaufen und durch eine Reihe an Fragen konnten wesentliche Informationen zur aktuellen Situation der anrufenden Person ermittelt und innerhalb der *Dataobjects* und *Attributes* hinterlegt werden. Kurz zusammengefasst handelt es sich bei der Anruferin um eine 35-jährige Frau, welche über starke Schmerzen in ihrer Brust klagt. Nach dem Vorbild der bSNA endet an dieser Stelle die Unterstützung der Disponent:innen, die nun entweder außerhalb des Systems noch weitere Fragen stellen oder direkt eine Entscheidung in Form einer geeignet erscheinenden Disposition vornehmen.

Abbildung 8.9 zeigt, wie sich die aktuelle Prozesssituation in Form einer aPV gegenüber den Anwender:innen darstellt. Hierbei ist das grün umrandete Segment ① von besonderer Bedeutung. Im Falle einer Ausführung ausschließlich nach bSNA wird das Segment ① nicht angezeigt und die Prozessinstanz verhält sich entsprechend der in der Cobra Leitstellen-Software bekannten Art und Weise. Falls die Ausführung mit den Erweiterungen der eSNA durchgeführt werden soll, wird das Segment dargestellt und die Anwender:innen erhalten gleich mehrere Optionen in Form von empfohlenen Fragen, die der anrufenden Person gestellt werden können.

Bei dem dargestellten Segment ① handelt es sich um eines der eingeführten KIS-Prozesselemente einschließlich der damit in Verbindung stehenden Fragen.

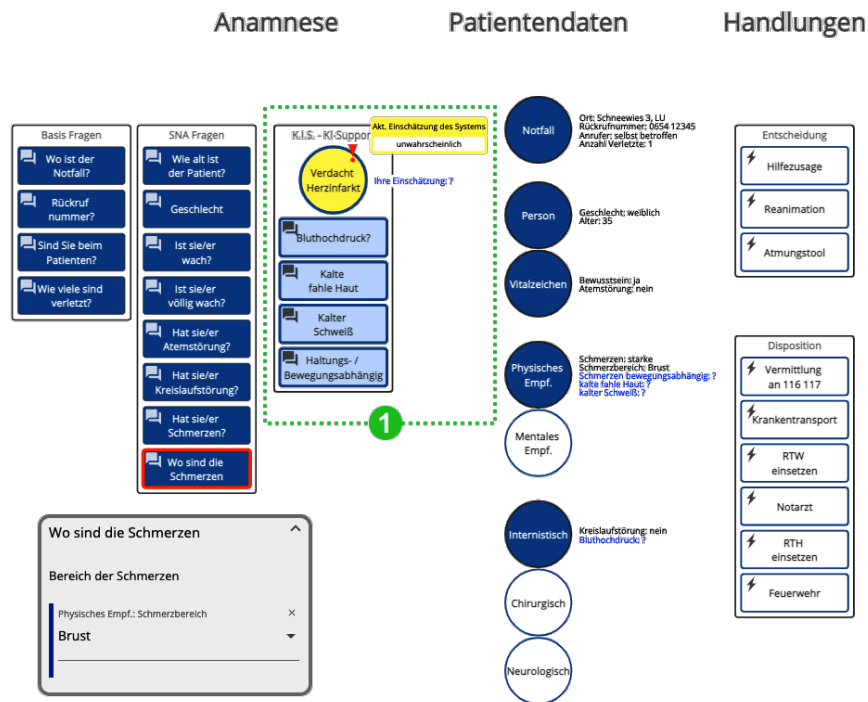


Abbildung 8.9: aPV einer Prozessinstanz nach Durchlauf gemäß bSNA

Die *Aktivierung* des KIS-Elements erfolgt durch das Expertensystem in Form der in der Domain-Ontologie hinterlegten Regeln. Kurz gesagt: Wenn die Prozessinstanz ein KIS-Element vom Typ „Verdacht Herzinfarkt“ besitzt und die Datenlage auf Schmerzen in der Brust hindeuten, dann wird das KIS-Element (genauer: das Attribut „Ihre Einschätzung“) als *Process-Goal* geschlussfolgert. Solange dieses Attribut nicht gesetzt ist (Placeholder), wird das Prozesselement auch als *goal-relevant* geschlussfolgert, was sich in der gelben Hervorhebung ausdrückt.

Die *Verbreitung* erfolgt nun über ein in der aPV nicht dargestelltes Taskelement „KA-Herzinfarkt“ (siehe Abbildung 8.7), worüber dieser *goal-relevant* Status an weitere Attribute-Elemente weitergegeben wird, welche nun neben den zugehörigen *Dataobjects* in blau angezeigt werden. Von dort überträgt sich der Status *goal-relevant* auf die zugehörigen Fragen-Elemente, welche nun ebenfalls angezeigt und unterhalb des KIS-Elementes

angeordnet werden.

Aufgrund der vorliegenden Daten und des zugrundeliegenden Expertensystems kommt das System entsprechend der durch den ärztlichen Leiter vorgegebenen Regeln zu einer ersten, vorläufigen *Einschätzung*. Ein Herzinfarkt ist demnach zunächst unwahrscheinlich, da dies bei Frauen in einem Alter unter 50 Jahren nur äußerst selten vorkommt. Mit der Beantwortung der vorgeschlagenen Fragen kommen im weiteren Verlauf dann zusätzliche Informationen hinzu, welche zu einer Veränderung dieser System-Einschätzung führen können.

Innerhalb des Segments ① können nun durch die Erweiterungen der eSNA die Vorteile des neuen ODD-BP Ansatzes genutzt werden, indem gleich mehrere Tasks gleichzeitig den Anwender:innen als Ausführungsoptionen angeboten werden. Die Reihenfolge, in der die Fragen gestellt werden, ist dabei unerheblich, während die jeweils im Prozess eingegeben Antworten unmittelbar berücksichtigt und gegebenenfalls zu einer Aktualisierung der System-Einschätzung führen. Mit diesem adaptiven und anwenderbezogenen Verhalten der eSNA durch die Prozessvisualisierung und der Integration des Expertenwissens wird erkennbar, wie die Disponent:innen aktiv bei der Abarbeitung des jeweiligen Notrufs unterstützt werden sollen.

8.4.3 Prozessinstanz nach eSNA

Innerhalb der exemplarische Prozessinstanz sind aufgrund der Fragen, motiviert durch das KIS-Element, weitere Daten ermittelt worden, welche zu einer Neubewertung der Systemeinschätzung führen. Anwender:innen können sich daran orientieren oder eine davon abweichende eigene Einschätzung vornehmen. Mit der Eingabe der eigenen Einschätzung geht eine Dispositionsempfehlung einher. Dieser können die Disponent:innen wieder folgen oder, wenn es angeraten scheint, davon abweichen.

Abbildung 8.10 zeigt die zum beschriebenen Prozesszustand passende Prozessdarstellung. Alle Fragen (*Tasks*) zum KIS-Element wurden durchgeführt und die als *goal-relevant* eingestuften Attribute wurden befüllt. In der aktuellen Darstellung ist erkennbar, dass das KIS-Element selbst selektiert ist und die Anwender:innen auf der rechten Seite ihre eigene Einschätzung getroffen haben. Mit der gelb hervorgehobenen Handlungsoption wird vermittelt, welche Disposition im Zusammenhang mit der Einschätzung empfohlen wird. Auch hier gilt wieder, dass diese Empfehlung sowohl das Ergebnis von Expertenwissen innerhalb der Domain-Ontologie sein kann als auch auch als Teil der Prozessdefinition modelliert werden kann. Im vorliegenden

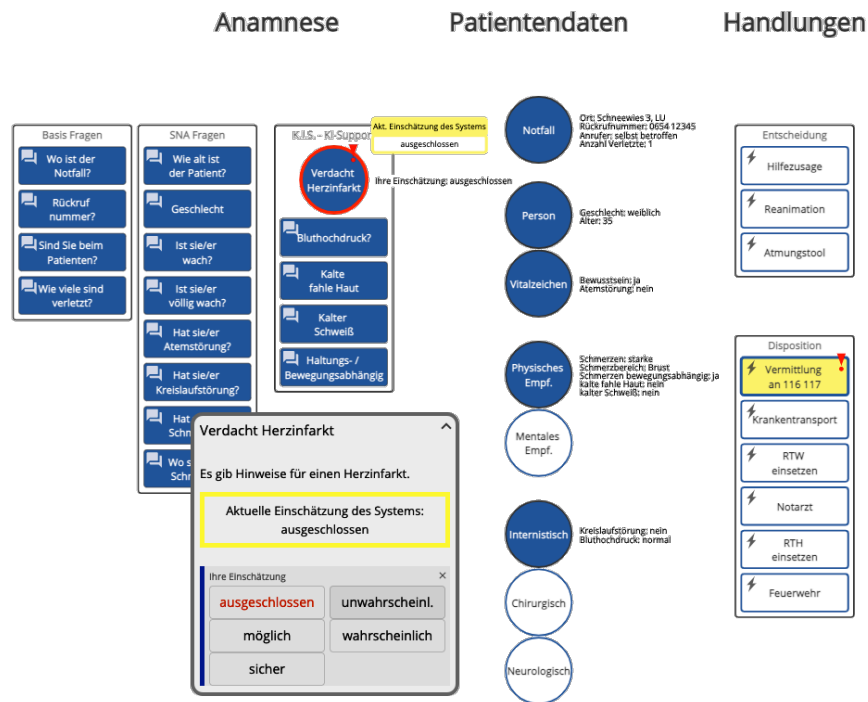


Abbildung 8.10: aPV einer Prozessinstanz nach Durchlauf inkl. eSNA

Fall wurde die zweite Option gewählt. Mit der Auswahl einer oder mehrerer Handlungsoptionen endet der für die Evaluation vorbereitete Rahmen der bSNA und eSNA.

Bei der Anpassung der generischen Basis an das Anwendungsszenario sind sowohl grundlegende Anforderungen der Notruf-Abfrage als auch Anforderungen hinsichtlich der durchzuführenden Evaluation berücksichtigt worden.

So ist von Seiten der ILtS-Ludwigshafen die Erwartung formuliert worden, dass die Anwender:innen aufgrund neuer Freiheitsgrade nicht mit einer zu komplexen Darstellung konfrontiert werden sollen. Das System soll die Nutzer:innen zu den wichtigen Fragen leiten, keine eigenen Entscheidungen treffen, jedoch Empfehlungen und Einschätzungen aussprechen. Hierdurch können die vom ärztlichen Leiter vorgegebenen und im Expertensystem umgesetzten Richtlinien zur Orientierung genutzt werden. Ziel ist es, auf diesem

Wege zu einer Verbesserung und gleichzeitig Vereinheitlichung der Dispositionsentscheidungen beizutragen.

Für die angestrebte Evaluation ist es darüber hinaus von Bedeutung, dass der Status-Quo mit einem neuen Verfahren verglichen werden kann. Zu diesem Zweck wurden beide Verfahren in eine gemeinsame Prozessdefinition integriert, sodass jederzeit festgelegt werden kann, ob ein Testfall nach bSNA oder nach eSNA durchgeführt werden soll. Folgt man ausschließlich den angebotenen Fragen und verzichtet auf das Anzeigen eines KIS-Elementes, so folgt der Prozess vollständig dem durch das Leitstellen-System bekannten bSNA-Ablauf. Erlaubt man hingegen den Anwender:innen jederzeit Daten abseits der Fragen innerhalb der Datenelemente einzugeben und zeigt je nach Datenlage auch die vorbereiteten KIS-Elemente an, so repräsentiert dies den eSNA Ablauf mit dem Ziel die Disponent:innen von den Vorteilen des ODD-BP Ansatzes profitieren zu lassen.

Nachfolgend wird im Rahmen der Evaluation auf diesem Systemaufbau aufgesetzt und somit das prototypische System zur Überprüfung von Hypothesen und zur Beantwortung der Forschungsfragen verwendet.

9 Evaluation

Im Zuge der Evaluation wird das Ziel verfolgt, Antworten auf die beiden initial gestellten Forschungsfragen zu erhalten. Hierzu werden Hypothesen formuliert, welche die Erwartungshaltung an den neu entworfenen ODD-BP Ansatz und die hierzu konzipierten Methoden und Verfahren widerspiegeln. Zur Bestätigung bzw. Widerlegung dieser Hypothesen wird das in Kapitel 8 vorgestellte prototypische System verwendet und mit Anwender:innen aus der Leitstellen-Domäne getestet. Hierzu wird ein experimenteller Aufbau zur Überprüfung der Hypothesen vorgestellt. Entsprechend dieses Setups wurde eine Testreihe mit 21 Anwender:innen der ILtS-Ludwigshafen durchgeführt. Mit den durch die Testreihen ermittelten Daten wird die Evaluation durchgeführt und anschließend zur Beantwortung der Forschungsfragen verwendet.

9.1 Hypothesen

Innerhalb der Einleitung (Kapitel 1.2) wurden die grundsätzlichen Anforderungen wissensintensiver Prozesse hinsichtlich Flexibilität, Adaptivität und Zielorientierung identifiziert. Im Anschluss wurden drei Ansätze vorgestellt, welche vorteilhafte Eigenschaften hinsichtlich dieser Anforderungen aufzeigen. Im Einzelnen sind dies der deklarative, der datenzentrierte und der semantische Prozessmodellierungsansatz. Hieraus leitete sich die Frage (*Forschungsfrage 1*) ab, ob diese drei bekannten Prinzipien der Prozessmodellierung in einen gemeinsamen und einheitlichen Ansatz überführt werden können und ob dieser dann die identifizierten Anforderungen wissensintensiver Prozesse erfüllt. Die in Kapitel 5, 6, 7 vorgestellten Methoden und Verfahren sind mit Blick auf diese Forschungsfrage entworfen und konzipiert worden, um den genannten Anforderungen gerecht zu werden. Mit dem in Kapitel 8 vorgestellten prototypischen System, das die neuen Methoden und Verfahren des ODD-BP Ansatzes umsetzt, wird die Erwartungshaltung verknüpft, dass sich deutliche Verbesserungen hinsichtlich der identifizierten Anforderungen bei KiPs erkennen lassen. Dies führt zu den ersten beiden Hypothesen.

Hypothese 1

Der ODD-BP Ansatz unterstützt eine flexible Prozessausführung, was zu einer höheren Variabilität ausgeführter Aktivitäten führt, im Vergleich zu dem etablierten und auf einem Kontrollfluss basierenden Verfahren.

Hypothese 2

Der ODD-BP Ansatz unterstützt eine adaptive und zielorientierte Prozessausführung, was sich in einer angepassten Prozessvisualisierung widerspiegelt, welche die Anwender:innen bei der Prozessausführung unterstützt.

Neben diesen Kernanforderungen wissensintensiver Prozesse ist angestrebt, dass verfügbares Wissen im Rahmen einer Prozessausführung genutzt werden könnte. Dies wird adressiert durch die Frage (Forschungsfrage 2), ob in einem einheitlichen Ansatz generisches Wissen so integriert werden kann, dass dieses in einer Prozessausführung aktiv einen Beitrag leisten und die Anwender:innen unterstützen kann. Hierzu wurde, abgestimmt auf das zur Evaluation festgelegte Anwendungsszenario, Expertenwissen als domainspezifische Erweiterung des generischen prototypischen Systems in den Prozesskontext eingebunden. Naturgemäß kann ein Beitrag zur Unterstützung der Anwender:innen auf unterschiedlichen Wegen geleistet werden und kann sich in unterschiedlichen quantitativen oder qualitativen Kenngrößen bemerkbar machen. Im Rahmen dieser Arbeit wird eine qualitative Verbesserung durch das integrierte Wissen angestrebt, was zur dritten Hypothese führt.

Hypothese 3

Mit dem ODD-BP Ansatz und unter Ausnutzung eingebetteten Expertenwissens kann eine qualitative Steigerung des Prozessergebnisses im Vergleich zu dem etablierten Verfahren erzielt werden.

Im Folgenden wird zunächst der experimentelle Aufbau vorgestellt und im Anschluss werden die einzelnen Hypothesen nochmals aufgegriffen, im Detail beschrieben und anschließend mit Hilfe der gewonnenen Testdaten bewertet.

9.2 Experimenteller Aufbau

Einleitend muss hervorgehoben werden, dass alle Aspekte mit einem direkten Bezug zum Anwendungsfall einer Notrufabfrage in enger Abstimmung mit dem ärztlichen Leiter und erfahrenen Disponent:innen der ILtS-Ludwigshafen ausgearbeitet wurden. Auf dieser Grundlage wurde bereits in Kapitel 4 der etablierte SNA-Ablauf untersucht und die bestehenden Defizite identifiziert. Auch Kapitel 8, die domainspezifische Anpassung des prototypischen Systems, basiert wesentlich auf dieser Zusammenarbeit. In gleicher Weise wurde auch mit Hilfe dieser fachlichen Expertise der gesamte experimentelle Aufbau gestaltet. Dies betrifft in besonderem Maße die nachfolgend vorgestellten, beispielhaften Notfälle, die einen klaren Praxisbezug aufweisen. Diese Notfälle wurden nicht nur hinsichtlich den vorbereiteten Fragen und entsprechenden Antworten aufgebaut, sondern es wurde vielmehr eine vollständige Hintergrund-Story zu jedem einzelnen Fall entwickelt, damit die teilweise individuellen Nachfragen der Disponent:innen, konform zum jeweiligen Fall, spontan beantwortet werden konnten. Zudem wurden die Notfälle so konzipiert, dass es je nach Detailgrad der Befragung zu unterschiedlichen Einschätzungen und somit zu unterschiedlichen Dispositionsergebnissen kommen kann.

Nachfolgend werden zunächst die drei simulierten Notfälle detailliert beschrieben und die jeweilige Interpretationen und die damit zusammenhängende und zu erwartende Dispositionen erläutert. Ergänzend werden die jeweiligen KIS-Elemente vorgestellt, einschließlich dem hierzu in der Domain-Ontologie hinterlegten Expertenwissen. Danach wird der generelle Ablauf eines Testlaufs dargelegt sowie die Umsetzung zur Protokollierung der Interaktionen und Entscheidungen der Anwender:innen. Zudem wird das grundlegende Schema der Testdurchläufe vorgestellt, was neben den drei Testfällen auch einen Fragenkatalog beinhaltet, der von den Teilnehmer:innen beantwortet wurde. Letztlich wird noch die Zusammensetzung der Testgruppe erörtert sowie das verwendete technische Setting beschrieben.

9.2.1 Simulierte Notfälle

Um ein möglichst breites Spektrum mit den simulierten Notfällen abdecken zu können, sind sehr unterschiedliche Fälle mit unterschiedlichen Hintergrundstories ausgearbeitet worden. Hierzu wurde ein Fall gewählt, der sich zunächst wie ein schwerwiegender Notfall präsentiert, jedoch bei einer detaillierten Betrachtung eher harmloser Natur ist. Ein weiterer Fall folgt einem inversen Verlauf. Nach einem zunächst harmlos klingenden Vorfall wird bei Kenntnis über weitere Details ein höherer Schweregrad erkennbar. In einem weiteren Fall ist die Schwere des Notfalls sofort eindeutig, allerdings ist die Ursache von Bedeutung und gleichzeitig schwer zu erkennen. Nachfolgend werden die drei Fälle mit A,B,C gekennzeichnet und im weiteren Verlauf wird mit diesen Bezeichnern immer wieder zu dem jeweiligen Fall referenziert.

Fall A: Bewusstlosigkeit

Im Fall A wird die bewusstlose Mutter von der Anruferin (Tochter) in der Wohnung vorgefunden. Die Startbeschreibung der Tochter zum Beginn des Notrufs lautet: „Hallo, ich habe meine Mutter ohnmächtig in ihrer Wohnung gefunden. Vor 30 Minuten hat sie mich noch angerufen und über Kopfschmerzen geklagt, daraufhin bin ich zu ihr gefahren.“

Im Rahmen der bSNA werden zu den zu erwartenden Fragen folgende Antworten gegeben.

- Wo ist der Notfall? In der Eichenstraße 2b in Ludwigshafen
- Wie ist Ihre Rückrufnummer? 0160/121212
- Sind Sie bei der betroffenen Person? Ja, ich bin bei ihr.
- Wieviele Betroffene gibt es? Eine - nur meine Mutter.
- Wie alt ist die betroffene Person? 68.
- Welches Geschlecht hat die betroffenen Person? Weiblich.
- Ist die Person bei Bewusstsein? Nein.
- Atmet die Person? Ja
- Seit wann ist die Person ohnmächtig? (gerade) Vor 30 Minuten haben wir telefoniert, sie klagte über Kopfschmerzen.

An dieser Stelle endet die durch die bSNA vorgegebenen Fragen. Im Falle einer bewusstlosen Person wird dann standardmäßig ein Rettungswagen mit

Notarzt alarmiert. Die Ursache der Bewusstlosigkeit ist noch unklar und die Disponent:innen sind auf sich gestellt, um weitere Details zu ermitteln. Somit bleibt gegebenenfalls unerkannt, dass die Ursache eine Kohlenmonoxid-Vergiftung aufgrund einer defekten Gastherme in der Wohnung (weit verbreitet in Ludwigshafen) ist. Einen ersten schwachen Hinweis hierzu gab es durch die Schilderung der Tochter, dass ihre Mutter zuvor über Kopfschmerzen klagte.

Für die eSNA wurde zu Fall A ein KIS-Element konzipiert, das im Falle einer Bewusstlosigkeit das Ziel verfolgt, die mögliche Ursache durch geeignete Fragen zu ermitteln. Vor einer Aktivierung des KIS-Elements wird jedoch die Disposition von zumindest einem Rettungswagen abgewartet, da nachfolgende Fragen nicht die Zeit bis zu einer ohnehin notwendigen Disposition verzögern sollen. Das generelle Verhalten des KIS-Elements wurde wie folgt innerhalb der Domain-Ontologie und der Prozessdefinition definiert:

- Aktivierung: Sobald ein Prozess ein KIS-Element vom Typ „Bewusstlos:Ursache?“ (*Uncertain_Cause_Unconscious*) besitzt und das Attribut „Bewusstsein“ (*PC_vital:consciousness*) den Wert *Nein* enthält und ein RTW (*disposition:rescue_vehicle*) disponiert wurde (Wert:*Einsatz*), dann wird das KIS-Element als *ProcessGoal* eingestuft.
- Verbreitung: Mit diesem KIS-Element sind die Attribute Diabetes (*PC_Internal:Diabetes*), Alkohol (*Person:Alcohol*), Medikamente (*Person:Medication*) und Gasofen/Gastherme (*Environment:GasHeating*) verknüpft. Entsprechend werden diese Elemente sowie die dazugehörigen Fragen als *goal-relevant* geschlussfolgert.
- Einschätzung: Jedes dieser Attribute kann eine Ursache für die Bewusstlosigkeit darstellen. Im vorliegenden Fall sind alle Einflussfaktoren auszuschließen, mit Ausnahme der Gastherme. Wird das entsprechende Attribut (*Environment:GasHeating*) auf den Wert Ja gesetzt, kommt das System zur Einschätzung, dass es sich um eine Hypoxie in Folge einer Gasvergiftung handeln könnte. Das KIS-Element wird der Klasse *Tag_Gasleak* zugeordnet, wodurch Anwender:innen entsprechende Hinweise dargestellt werden.

Sobald die Disponent:innen sich dieser Einschätzung anschließen, indem sie das KIS-Element mit einem der möglichen Werte *Gasvergiftung* oder *Rauchvergiftung* belegen, wird innerhalb der eSNA die Disposition der Feuerwehr und eine Handlungsempfehlung zum Öffnen der Fenster als *ProcessGoal* eingestuft.

Fall B: Schmerzen Brustbereich

Im Fall B ist die Anruferin selbst betroffen und klagt über starke Schmerzen in der Brust. Die Startbeschreibung zum Beginn des Notrufs lautet: „Hallo, ich habe starke Schmerzen in der Brust. Gestern war ich noch im Urlaub und wir sind von Südspanien mit dem Auto nach Hause gefahren und seit heute morgen habe ich diese Schmerzen.“

Im Rahmen der bSNA werden zu den vordefinierten Fragen folgende Antworten gegeben.

- Wo ist der Notfall? In der Eichenstraße 2b in Ludwigshafen
- Wie ist Ihre Rückrufnummer? 0160/121212
- Sind Sie bei der betroffenen Person? Ich bin selbst betroffen.
- Wieviele Betroffene gibt es? Eine - nur mich.
- Wie alt ist die betroffene Person? 29.
- Welches Geschlecht hat die betroffenen Person? Weiblich.
- Ist die Person bei Bewusstsein? Ja.
- Ist sie völlig wach und ansprechbar? Ja.
- Hat die Person Atemstörungen? Leichte.
- Hat die Person Kreislaufstörungen? Nein
- Hat die Person Schmerzen? Ja, starke Schmerzen.
- Wo sind die Schmerzen? Im Brustbereich.

Dies entspricht soweit dem durch die bSNA vorgegebenem Weg. Im Falle von starken Schmerzen in der Brust steht der Verdacht eines Herzinfarktes im Raum. Allerdings sprechen viele Indizien dagegen. So ist ein Herzinfarkt bei weiblichen Personen unter 50 Jahren äußerst selten. Zudem hat die Anruferin geschildert, dass sie gestern noch eine sehr lange und weite Autofahrt erlebt hat. Die Ursache der Schmerzen im Brustbereich, eine Intercostalneuralgie bei Muskelverspannungen nach einer langen Autofahrt, werden von den Fragen der bSNA jedoch nicht abgedeckt. Entsprechend bleibt es entweder bei dem initialen Verdacht auf einen Herzinfarkt und ein RTW wird entsendet oder die Disponent:innen fragen eigenständig nach weiteren Details zur besseren Beurteilung der Situation.

Für die eSNA wurde zu Fall B ein KIS-Element entworfen, um den Schweregrad des vorliegenden Notfalls einschätzen zu können. Die Aktivierung, Verbreitung und Einschätzung wird an dieser Stelle nicht nochmals wiederholt, da dies bereits in Kapitel 8.2.2 detailliert als Beispiel der KI-Support Konzepte und Regeln erfolgt ist. Durch das KIS-Element werden dabei erwartungsgemäß die möglichen Ursachen der Schmerzen in der Brust abgefragt und alle für einen Herzinfarkt sprechende Hinweise bleiben aus. Stattdessen wird die Frage, ob die Schmerzen bewegungsabhängig sind, sich also je nach Körperhaltung verändern oder teilweise verschwinden, mit ja beantwortet. Dies wiederum ist ein klarer Hinweis darauf, dass die Schmerzen in einem Zusammenhang mit dem Bewegungsapparat stehen und es sich somit nicht um einen Herzinfarkt handeln kann. Mit Fall B wird demnach mit einem schwerwiegenden Verdacht gestartet, im Verlauf der Befragung, speziell innerhalb der eSNA, verdichten sich jedoch die Hinweise, dass den Schmerzen eine eher harmlose Ursache zugrunde liegt. Entsprechend fällt die System-Einschätzung aus und wenn die Disponent:innen sich dieser Einschätzung anschließen, wird auch lediglich eine Vermittlung an die ärztliche Hotline 116 117 empfohlen. Wird von dieser Empfehlung abgewichen, wird, je nach eingeschätzter Wahrscheinlichkeit eines Herzinfarktes, eine abgestufte Disposition von Krankentransport über RTW bis hin zu einem Notarzt empfohlen.

Fall C: Verkehrsunfall

Im Fall C meldet sich eine Anruferin hinsichtlich eines augenscheinlich harmlosen Verkehrsunfalls. Die Startbeschreibung der Anruferin lautet: „Hallo, ich habe gerade beim Herausfahren aus einem Parkplatz einen älteren Mann mit dem Auto umgestossen. Dieser sagt nun, er kann nicht mehr aufstehen.“

Im Rahmen der bSNA werden zu den vordefinierten Fragen folgende Antworten gegeben.

- Wo ist der Notfall? In der Eichenstraße 2b in Ludwigshafen
- Wie ist Ihre Rückrufnummer? 0160/121212
- Sind Sie bei der betroffenen Person? Ja, ich bin vor Ort.
- Wieviele Betroffene gibt es? Eine - der ältere Herr.
- Wie alt ist die betroffene Person? vermutlich 80
- Welches Geschlecht hat die betroffenen Person? Männlich.
- Ist die Person bei Bewusstsein? Ja.

- Ist sie völlig wach und ansprechbar? Ja.
- Hat die Person Atemstörungen? Der Mann sagt nein.
- Hat die Person Kreislaufstörungen? Kann ich nicht sagen, Unklar.
- Hat die Person Schmerzen? Ja, leichte Schmerzen.
- Wo sind die Schmerzen? Im Beckenbereich.

Die Fragen der bSNA hinsichtlich Fall C enden an dieser Stelle. Unklar bleibt zunächst, wie schwerwiegend die Verletzung des älteren Mannes einzuschätzen sind. Die Beschreibungen der Anruferin sind eher beschwichtigend und lassen den Vorfall eher „klein“ erscheinen. In diese Unsicherheit hinein wird nun eine Dispositionsentscheidung getroffen, falls die Disponent:innen nicht aus eigenem Antrieb heraus noch weitere Fragen abseits der bSNA stellen.

Für die eSNA wurde auch für Fall C ein KIS-Element konzipiert, welches das Ziel verfolgt, den Schweregrad der Verletzung besser beurteilen zu können. Dies wurde innerhalb der Domain-Ontologie und der Prozessdefinition wie folgt umgesetzt:

- Aktivierung: Sobald ein Prozess ein KIS-Element vom Typ „Schweregrad Verletzung“ (*Severity_Level_Fall*) besitzt und Schmerzen in einem lebenswichtigen Körperbereich vorkommen (*PC_physical:pain_location* mit einem Wert *Kopf, Nacken, Rücken, Becken, Hüfte, Oberschenkel*), dann wird das KIS-Element als *ProcessGoal* eingestuft.
- Verbreitung: Mit diesem KIS-Element sind die Attribute Kalte-fahle-Haut (*PC_physical:cold_fade_skin*), Blutungen (*PC_surgical:bleeding*) und Frakturen (*PC_surgical:fracture*) verknüpft. Entsprechend werden diese Elemente sowie die dazugehörigen Fragen als *goal-relevant* geschlossen.
- Einschätzung: Jedes dieser Attribute kann als Hinweis für eine schwerwiegende Verletzung sein. Im vorliegenden Fall wird das Attribut Kalte-fahle-Haut mit Ja bestätigt, was ein starkes Indiz auf ein Kreislaufproblem ist. Entsprechend wird die System-Einschätzung hinsichtlich des Schweregrades als kritisch eingestuft.

Schließen sich die Disponent:innen dieser Einschätzung an, wird als Dispositionsempfehlung RTW und Notarzt hervorgehoben. Bei einer anderen Einstufung (harmlos, mittel, ernst) wird eine entsprechend abgestufte Empfehlung der Einsatzmittel vorgeschlagen.

- Aktivierung: Sobald ein Prozess ein KIS-Element vom Typ „Schweregrad Verletzung“ (*Severity_Level_Fall*) besitzt und Schmerzen in einem lebenswichtigen Körperbereich vorkommen (*PC_physical:pain_location* mit einem Wert *Kopf, Nacken, Rücken, Becken, Hüfte, Oberschenkel*), dann wird das KIS-Element als *ProcessGoal* eingestuft.
- Verbreitung: Mit diesem KIS-Element sind die Attribute Kalte-fahle-Haut (*PC_physical:cold_fade_skin*), Blutungen (*PC_surgical:bleeding*) und Frakturen (*PC_surgical:fracture*) verknüpft. Entsprechend werden diese Elemente sowie die dazugehörigen Fragen als *goal-relevant* geschlossenfolgert.
- Einschätzung: Jedes dieser Attribute kann als Hinweis für eine schwerwiegende Verletzung sein. Im vorliegenden Fall wird das Attribut Kalte-fahle-Haut mit Ja bestätigt, was ein starkes Indiz auf ein Kreislaufproblem ist. Entsprechend wird die System-Einschätzung hinsichtlich des Schweregrades als kritisch eingestuft.

9.2.2 Fragenkatalog

Neben den vorbereiteten Notfällen wurde zusätzlich ein Fragenkatalog vorbereitet, womit zum Ende der drei simulierten Testfälle die Eindrücke der Teilnehmer:innen zu verschiedenen Aspekten des Systems abgefragt wurde. Die gestellten Fragen lauteten:

- Bedienung
 1. Nachvollziehbarkeit: Konnten Sie die Bedienung im neuen System gut nachvollziehen?
 2. Umstieg: Wie leicht fiel Ihnen der Umstieg von Cobra auf diese neue Oberfläche?
 3. Erweiterung KIS: Wie hilfreich empfanden Sie den KI-Support als Erweiterung zur klassischen SNA?
 4. Umgang KIS: Wie leicht empfanden Sie den Umgang mit dem KI-Support auf Basis des Expertensystems?
 5. Datenelemente: Wie nützlich war es für Sie, dass Daten jederzeit und abseits von Fragen eingegeben werden konnten?
- Darstellung
 6. Anordnung: Wie übersichtlich empfanden Sie die Anordnung von Fragen, Patientendaten und Handlungsoptionen?

7. Patientendaten: Wie hilfreich war die Ansammlung von Patientendaten in der mittleren Spalte?
 8. Formen: War die Unterscheidung der Elemente in Daten (Kreis) und Fragen/Handlungen (Rechtecke) für Sie gut verständlich?
 9. Farben: Die Farbe der Elemente gibt Auskunft über den Status und die Dringlichkeit. War dies für Sie gut nachvollziehbar?
 10. Dateneingabe: Wie benutzerfreundlich war für Sie die Eingabe von Daten auf der rechten Seite?
- KIS
 11. Inhalte: K.I.S. steht für KI-Support. Wie passend empfanden Sie die vorgeschlagenen Fragen in der jeweiligen Situation?
 12. Systemeinschätzung: Nachdem die Fragen beantwortet waren: Wie gut entsprach die Einschätzung des Systems auch Ihrer Beurteilung der Situation?
 13. Dispositionsempfehlungen: Wie passend empfanden Sie die Empfehlungen zur Disposition?
 14. Unterstützung: Wie gut denken Sie, dass Sie und Ihre Kollegen mit einem solchen System bei Ihrer täglichen Arbeit unterstützt würden?
 15. Wertung: Würden Sie den Einsatz eines KI-Supports - wie hier getestet - für die Leitstelle begrüßen?

Die Bewertung zu jeder der gestellten Fragen erfolgte nach dem vertrauten Notensystem aus der Schule. Eine 1 steht für eine sehr gute bzw. positive Bewertung des jeweiligen Aspekts, während eine 6 für die schlechtest mögliche Bewertung steht. Abschließend hatten die Teilnehmer:innen in einem freien Textfeld noch die Gelegenheit, Vorschläge für Verbesserungen einzugeben.

9.2.3 Ablauf eines Testdurchlaufs

Die drei vorbereiteten Testfälle wurden in einem möglichst der Praxis entsprechenden Rahmen durchgeführt. Hierzu hat eine Anruferin eine vorbereitete Nummer der ILtS-Ludwigshafen verwendet, um wie ein regulärer Notruf in die Notruf-Zentrale geschaltet, dort jedoch an einen zuvor festgelegten Arbeitsplatz vermittelt zu werden. Somit wurde die gleiche technische Ausstattung wie bei einem normalen Notruf verwendet, einschließlich der Telefonanlage und der damit einhergehenden Steuerung. Ferner wurde

mit dem gleichen Computer mit bekanntem Setting der Monitore, Maus und Tastatur gearbeitet. Statt der Cobra-Leitstellen-Software wurde dann lediglich mit einem Browser die Client-Server-Lösung der prototypischen Umsetzung von ODD-BP genutzt.

Jede am Test teilnehmende Person hat alle drei Testfälle durchgespielt und anschließend den vorgestellten Fragenkatalog beantwortet. Da im Vorfeld die Teilnehmer:innen nicht geschult werden konnten, wurde vor dem ersten Fall die Darstellung mit dem Aufbau der Elemente vorgestellt und der grundsätzliche Umgang mit dem System erklärt.

Innerhalb des ersten Testfalls wurde rein nach dem bSNA-Ablauf gearbeitet und den Anwender:innen wurde lediglich die Interaktion mit den dargestellten Fragen und Dispositionen ermöglicht. Entsprechend wurde auch zu keinem Zeitpunkt ein KIS-Element dargestellt. So konnte sichergestellt werden, dass der erste durchgeführte Fall nur dem etablierten Ablauf der durch das Cobra-Leitstellensystems bekannten Notrufabfrage entspricht.

Im Anschluss wurde im zweiten Testfall nach den Fragen anhand des bSNA-Ablaufs das jeweils vorbereitete KIS-Element angezeigt und der damit verbundene eSNA-Ablauf erweiterte das bislang bekannte Frage-Spektrum. Zudem wurde, nachdem die Disponent:innen ihre jeweilige eigene Einschätzung zum KIS-Element festgelegt hatten, eine passende Handlungsempfehlung vorgeschlagen, an der man sich orientieren konnte jedoch nicht daran halten musste.

Der dritte Testfall verhielt sich grundsätzlich wie der Zweite, allerdings wurden die Anwender:innen nun gezielt darauf aufmerksam gemacht, dass sie jederzeit auch Daten frei über die Datenelemente und den darin enthaltenen Attributen eingeben können. Dies zielte vor allem auf die jeweiligen Startaussagen zu den einzelnen Fällen ab, welche bereits Informationen enthalten, die ansonsten erst sehr viel später im Verlauf der bSNA abgefragt werden.

Um den bSNA- und eSNA-Ablauf bestmöglich miteinander vergleichen zu können, wurde jeder der Testfälle (A,B,C) abwechselnd als ersten, zweiten oder dritten Testfall durchgespielt. So konnte für jeden der vorbereiteten Fälle dokumentiert werden, wie sich die Anwender:innen im jeweiligen Testverlauf verhalten und welches Dispositionsergebnis hieraus resultiert. Die Protokollierung erfolgte dabei nicht nur über die Wissensbasis selbst, sondern parallel über eine Protokollierung jeder einzelnen Nutzerinteraktion mit dem System. Mit Hilfe dieser Logbücher kann nicht nur jede Datenein-

gabe nachvollzogen werden, zudem kann durch exakte Zeitstempel auch die Dauer zwischen einzelnen Interaktionen analysiert werden.

Eine Disposition im Zuge einer Notrufabwicklung ist eine Entscheidung, welche auf Basis der verfügbaren Daten und den im Verlauf eines Notrufs gewonnenen Eindrücke beruht. Grundsätzlich gibt es hierzu von Seiten der in jeder Leitstelle zuständigen ärztlichen Leitung Vorgaben, in welchen Situationen welche Einsatzkräfte eingesetzt werden sollen. Diese Vorgaben werden den Disponent:innen überwiegend in Form von Richtlinien zur Verfügung gestellt und es bedarf einer detaillierten Kenntnis, um diese in der Praxis umzusetzen. Aus dieser Perspektive gibt es für jeden Notfall eine optimal geeignete Disposition von Einsatzkräften. In der Praxis sind die Dispositionen von vielen weiteren Faktoren abhängig, sie unterscheiden sich beispielsweise zwischen verschiedenen Mitarbeiter:innen und auch eine einzelne Person entscheidet nicht immer konstant.

Um die Qualität der Prozessergebnisse für die einzelnen simulierten Notfälle bewerten zu können, wurde durch den ärztlichen Leiter der ILtS-Ludwigshafen für jeden der drei vorbereiteten Fälle (A,B,C) eine optimale Disposition definiert. Hierzu wurden sieben unterschiedliche Dispositionsmöglichkeiten festgelegt: die Weiterleitung an die Hotline 116117, ein Krankentransport, ein Rettungstransport, ein Notarzt, ein Rettungshubschrauber, die Feuerwehr sowie eine gezielte Handlungsempfehlungen für die anrufende Person. Entspricht eine dieser Möglichkeiten dem jeweils erwarteten Ergebnis, wird dies als ein Punkt bewertet, bei einer Abweichung wird kein Punkt vergeben. So kann jedes Prozessergebnisse in Form eines Punktwertes zum Ausdruck gebracht werden, wobei der Punktwert sieben einem optimalen Ergebnis entspricht.

9.2.4 Teilnehmer:innen

Insgesamt wurden die drei Testfälle mit 21 Disponent:innen der ILtS-Ludwigshafen im Verlauf einer einzigen Woche durchgeführt. Die Gruppe bestand aus 16 männlichen und 5 weiblichen Personen mit einer Leitstellen-Erfahrung zwischen 0-28 Jahren (Durchschnitt 9). Im Verlauf von durchschnittlich 45 Minuten bekamen die Teilnehmer:innen eine kurze Einführung zum Umgang mit dem System und im Anschluss wurden die drei Testfälle durchgespielt. Im Ergebnis resultierte dies in 63 durchgeführten und aufgezeichneten simulierten Notruf-Abläufen.

Person	1.	2.	3.	Geschl.	Erfahrung	Organ.	Ausbildung
1	A	B	C	m	4	HO	Notfallsan.
2	A	C	B	m	7	HO	Rettungsassist.
3	B	A	C	m	8	HO	Notfallsan.
4	B	C	A	m	2	HO	Notfallsan.
5	C	A	B	m	11	HO	Notfallsan.
6	C	B	A	m	28	HO	Notfallsan.
7	A	B	C	m	15	HO	Notfallsan.
8	A	C	B	w	7	HO	Notfallsan.
8	B	A	C	m	3	HO	Notfallsan.
10	B	C	A	w	8	HO	Notfallsan.
11	C	A	B	m	8	BF	Rettungsassist.
12	C	B	A	m	11	HO	Notfallsan.
13	A	B	C	m	15	HO	Notfallsan.
14	A	C	B	w	1	BF	Rettungsassist.
15	B	A	C	m	26	HO	Rettungsassist.
16	B	C	A	m	15	BF	Rettungsassist.
17	C	A	B	w	0	HO	Notfallsan.
18	C	B	A	m	10	BF	Rettungsassist.
19	A	B	C	m	2	BF	Rettungsassist.
20	A	C	B	m	6	HO	Notfallsan.
21	B	A	C	w	1	HO	Notfallsan.

Tabelle 9.1: Übersicht der Testteilnehmer:innen inkl. Testpattern und individuellem Profil

Tabelle 9.1 zeigt eine Übersicht zu den einzelnen Testteilnehmer:innen einschließlich der Verteilung der drei vorbereitenden Fälle (A,B,C) hinsichtlich ihrer Verteilung als ersten, zweiten oder dritten Testfall. Darüber hinaus gibt die Tabelle Auskunft über das Geschlecht der einzelnen Testperson, ihre Leitstellenerfahrung in Jahren, der zugehörigen Organisation (HO=Hilfsorganisation, BF=Berufsfeuerwehr) sowie zum jeweiligen Ausbildungsstand (Notfallsanitäter, Rettungsassistent).

9.3 Ergebnisse zu Hypothese 1

Der ODD-BP Ansatz unterstützt eine flexible Prozessausführung, was zu einer höheren Variabilität ausgeführter Aktivitäten führt, im Vergleich zu dem etablierten und auf einem kontrollflussbasierenden Verfahren.

Die erste Hypothese adressiert die Vorteile einer flexiblen Prozessausführung, was sich in einer gesteigerten Variabilität widerspiegelt und sich darüber auch quantifizieren lässt. Hierzu müssen einleitend die Begrifflichkeiten hinsichtlich Flexibilität und Variabilität definiert werden.

Flexibilität wird häufig als eine wesentliche Fähigkeit, insbesondere bei wissensintensiven Prozessen, beschrieben. Jedoch konnte keine etablierte Methodik identifiziert werden, nach der Flexibilität in einem Prozess quantifiziert werden kann. Insbesondere auch mit Blick auf den ODD-BP Ansatz kann nicht eine einzige Zahl für einen ganzen Prozess berechnet werden, da ein Prozess nicht auf klassische Weise vordefiniert ist, sondern aus einem permanenten Prozessplanungsverfahren resultiert. Alternativ kann man sich die Flexibilität (Flex) als die Anzahl der Ausführungsoptionen (ExOpts) vorstellen, die bei einem bestimmten Prozessschritt (S) angeboten werden.

Während Flexibilität die Anzahl der Ausführungsoptionen für einen Prozessschritt ausdrückt, wird auch eine Terminologie benötigt, um die Anzahl der Nutzung dieser Flexibilität über eine Menge von Prozessausführungen auszudrücken. Zu diesem Zweck verwenden wir den Begriff Variabilität (Var), um die Intensität auszudrücken, in der die Anwender:innen die angebotene Flexibilität genutzt haben. Var_S hat einen Bereich zwischen 1, wenn keine Variationen aufgetreten sind, und $Flex(S)$ als Maximum, wenn alle Ausführungsoptionen gleichmäßig über eine Menge von ausgeführten Prozessen verwendet wurden. Dies führt zur Definition:

$$Var_S = \sum_{i=1}^{|ExOpts(S)|} \frac{numOccur(ExOpt_i, S)}{maxOccur(S)} \quad (R\ 25)$$

Formel R 25 definiert die durchschnittliche Variabilität eines bestimmten Prozessschrittes Var_S über eine Menge von durchgeführten Prozessinstanzen.

- $|ExOpts(S)|$ liefert die Anzahl unterschiedlicher Ausführungsmöglichkeiten zu einem bestimmten Prozessschritt S .
- $ExOpt_i$ repräsentiert eine spezifische Ausführungsoption.
- $numOccur(ExOpt_i, S)$ liefert die Anzahl an Ausführungen von $ExOpt_i$ zu einem bestimmten Prozessschritt S .
- $maxOccur(S)$ liefert die Anzahl des am häufigsten durchgeführten ExOpt für einen Prozessschritt S .

Abbildung 9.1 zeigt die gemessene, durchschnittliche Variabilität für alle Prozessschritte und alle Fälle. Jedes Diagramm zeigt die Ausführungen für jeweils einen der vorbereiteten Fälle (A,B,C) an. Die nachfolgende Zahl kennzeichnet, ob der jeweilige Fall im ersten, zweiten oder dritten Testdurchlauf durchgespielt worden ist. Wie im experimentellen Aufbau dargestellt, sind die Testfälle im ersten Testdurchlauf ausschließlich nach dem etablierten bSNA-Verfahren ausgeführt worden. Entsprechend wurde jeweils bei A1, B1 und C1 von Systemseite keine Flexibilität angeboten, weshalb die gemessene Variabilität genau 1 beträgt.

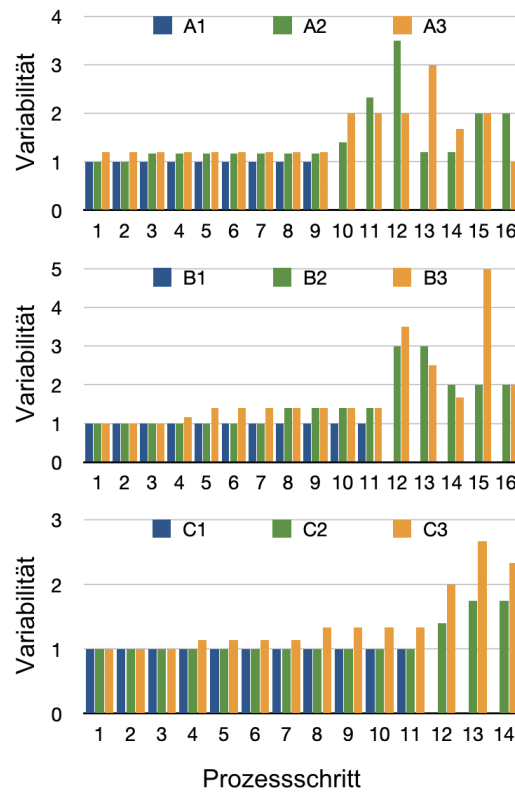


Abbildung 9.1: Gemessene durchschnittliche Variabilität $Var(S)$ für jeden Prozessschritt S

In den Experimenten, in denen die Fälle mit dem erweiterten eSNA-Verfahren ausgeführt wurden (A2, A3, B2, B3, C2, C3), nutzten die Anwender:innen die angebotene Flexibilität bis zu einem gewissen Grad, was zu der jeweils höheren durchschnittlichen Variabilität führt. Es ist erwähnenswert, dass im zweiten und dritten Testdurchlauf die Schritte des bSNA-Abschnittes (1-9 für Fall A und 1-11 für Fall B und C) ebenfalls Flexibilität ermöglichten, diese aber von den Anwender:innen kaum in Anspruch genommen wurde. In den einzelnen Fällen, in denen sie in Anspruch genommen wurde, haben die Disponent:innen die Informationen aus dem Eröffnungsstatement zu Beginn jedes Telefonats genutzt und diese gezielt in das Sys-

tem eingegeben. Überwiegend wurde jedoch dem bekannten und eintrainierten bSNA-Ablauf gefolgt.

Mit dem Verlassen dieses bekannten Bereichs und mit dem Erscheinen der vorbereiteten KIS-Elemente (10 bis Ende für Fall A und 12 bis Ende für Fall B und C) wurden neue und für die Anwender:innen unerwartete Prozesselemente angeboten. Die damit einhergehende Flexibilität wurde nun außerhalb des eintrainierten Pfades wesentlich stärker ausgenutzt. Infolgedessen verwandelte sich die angebotene Flexibilität in eine höhere, messbare Variabilität.

Die Auswertung der aufgenommenen Prozessinstanzen verifiziert die erste Hypothese und bestätigt, dass der ODD-BP Ansatz im Vergleich zum State-of-the-Art-Ansatz eine flexible Prozessausführung unterstützt.

9.4 Ergebnisse zu Hypothese 2

Der ODD-BP Ansatz unterstützt eine adaptive und zielorientierte Prozessausführung, was sich in einer angepassten Prozessvisualisierung widerspiegelt, welche die Anwender:innen bei der Prozessausführung unterstützt.

Die zweite Hypothese adressiert die Kernanforderungen hinsichtlich Adaptivität und Zielorientierung, welche mit Blick auf den OBB-BP Ansatz aus zwei Perspektiven betrachtet werden müssen. Zum einen können sich beide Begriffe auf das Prozessmodell beziehen, zum anderen kann es auf eine individuelle Prozessvisualisierung für eine einzelne Person Bezug nehmen.

Adaptivität wird klassisch, wie in Kapitel 2 eingeführt, innerhalb des Forschungsbereichs der Geschäftsprozessmodellierung verstanden als die Anpassung eines Prozessmodells an neue Anforderungen. Aufgrund der deklarativen Prozessmodellierung innerhalb des ODD-BP Ansatzes ist diese Form der Adaption nicht erforderlich, beziehungsweise wird im Rahmen dieser Arbeit nicht adressiert. Im Gegensatz dazu bezieht sich die Adaption der Prozessvisualisierung, so wie in Kapitel 7 vorgestellt, auf die für Anwender:innen maßgeschneiderte Anzeige einer Prozessinstanz. Im Zuge dieser Adaption werden Elemente ausgeblendet, aggregiert oder in die aPV aufgenommen, ohne dass das zugrundeliegende Prozessmodell verändert wird. Aus Sicht der Anwender:innen kann jedoch nicht differenziert werden zwischen einer Veränderung des Prozessmodells und einer Veränderung der Prozessvisualisierung.

Ähnlich verhält es sich mit der Zielorientierung. Die Identifikation neuer oder zusätzlicher Prozessziele kann grundsätzlich auch durch eine Anpassung des Prozessmodells umgesetzt werden. Dies ist innerhalb des ODD-BP Ansatz ebenfalls nicht erforderlich. Stattdessen wird die Zielorientierung durch die Einstufung der Relevanz der einzelnen Prozesselemente realisiert. Diese Relevanz bildet in der Folge wiederum eine zentrale Grundlage für die durchzuführende Adaption der Prozessvisualisierung.

Somit spiegeln sich letztlich sowohl die Adaption als auch die Zielorientierung in der adaptiven Prozessvisualisierung wider. In Folge dessen beruht die Evaluation dieser Hypothese auf empirischen Daten, erfasst durch die Befragung der Anwender:innen im Anschluss an die drei durchgeführten simulierten Notfälle. Nachfolgend werden die vollständigen Ergebnisse dieser Befragung dargestellt und anschließend mit Blick auf die zu untersuchende Hypothese interpretiert.

Frage	1	2	3	4	5	6	Mittelwert
Bedienung							
1. Nachvollziehbarkeit	11	9	1				1.52
2. Umstieg	4	10	6	1			2.19
3. Erweiterung KIS	8	10	2	1			1.81
4. Umgang KIS	8	10	1	2			1.86
5. Datenelemente	15	4	0	2			1.48
Darstellung							
6. Anordnung	1	12	7	1			2.38
7. Patientendaten	3	13	4	1			2.14
8. Formen	6	11	3	1			1.95
9. Farben	7	12	2				1.76
10. Dateneingabe	6	4	6	3	2		2.57
KIS-Elemente							
11. Inhalte	10	9	2				1.62
12. Systemeinschätzung	4	13	3	1			2.05
13. Dispositionsempfehlungen	9	11	1				1.62
14. Unterstützung	12	8	1				1.48
15. Wertung	15	5	1				1.33

Tabelle 9.2: Anzahl der jeweiligen Benotung zu jeder Frage inkl. Mittelwert

Insgesamt konnte im Rahmen der Teststudie eine sehr positive und aufgeschlossene Haltung gegenüber dem neuen ODD-BP Ansatz festgestellt werden. Dies wird durch eine generell sehr gute Bewertung zu den meisten der gestellten Fragen sichtbar. Irritationen wurden vor allem durch Aspekte verursacht, welche im Gegensatz zu den Gewohnheiten aus dem etablierten Cobra-Leitstellen-System standen. So musste beispielsweise eine Dateneingabe im Testsystem mittels Return abgeschlossen werden, während die Anwender:innen hierzu die Tab-Taste gewöhnt sind. Dies spiegelt sich in den Ergebnissen zu Frage 2, 6 und 10 wider.

Grundsätzlich ist es komplex, Funktionalitäten empirisch zu evaluieren, welche als solche für die Anwender:innen nicht sichtbar werden, sondern eher ambient in einem System integriert sind. Nach der Zufriedenheit kann in diesen Fällen nicht gezielt gefragt werden, ein Umstand der sowohl auf die aPV als auch auf die Zielorientierung zutrifft. Die Beurteilung kann dann lediglich durch die generelle Akzeptanz sowie aufgrund der Bewertung der durch die Funktionalitäten wesentlich beeinflussten Charakteristiken des Testsystems erfolgen. Nachfolgend werden gezielt die Fragen und ihre Bewertungen aufgegriffen, welche in einem Zusammenhang mit diesen durch die Funktionalitäten beeinflussten Charakteristiken betrachtet werden können.

F1 | *Nachvollziehbarkeit:* „Konnten Sie die Bedienung im neuen System gut nachvollziehen?“

Mit dieser Frage wird die grundsätzliche Plausibilität der aPV für die Anwender:innen adressiert. Diese kann als sehr gut eingestuft werden, da mit 1.52 eine der besten Bewertungen erzielt werden konnte.

F3 | Erweiterung KIS: „Wie hilfreich empfanden Sie den KI-Support als Erweiterung zur klassischen SNA?“

F4 | Umgang KIS: „Wie leicht empfanden Sie den Umgang mit K.I.S. - dem KI-Support auf Basis des Expertensystems?“

Diese beiden Fragen fokussieren die KI-Support-Elemente, welche wesentlich auf dem Charakter der Zielorientierung basieren sowie die Möglichkeiten einer aPV ausnutzen. Mit 1.81 und 1.86 wurde auch dies von den Disponent:innen mit einer guten Bewertung versehen.

F14 Unterstützung: Wie gut denken Sie, dass Sie und Ihre Kollegen mit einem solchen System bei Ihrer täglichen Arbeit unterstützt würden?

F15 Wertung: Würden Sie den Einsatz eines KI-Supports - wie hier getestet - für die Leitstelle begrüßen?

Die Bewertung beider Fragen repräsentiert eine generelle Offenheit und eine grundlegende Akzeptanz des evaluierten Testsystems einschließlich des zugrundeliegenden Expertensystems. Mit 1.48 und 1.33 wurden über alle Fragen hinweg die besten Ergebnisse erzielt.

Grundsätzlich kann eine hohe Akzeptanz und Offenheit gegenüber dem neuen Ansatz festgestellt werden und insbesondere die durch die aPV und Zielorientierung wesentlich beeinflussten Charakteristiken des Testsystems wurden sehr positiv bewertet. Auch wenn ein klarer Beweis auf diesem Wege für *Hypothese 2* nicht abgeleitet werden kann, so kann mit den Ergebnissen diese zumindest als *sehr plausibel* eingestuft werden.

9.5 Ergebnisse zu Hypothese 3

Mit dem ODD-BP Ansatz und unter Ausnutzung eingebetteten Expertenwissens kann eine qualitative Steigerung des Prozessergebnisses im Vergleich zu dem etablierten Verfahren erzielt werden.

Die dritten Hypothese adressiert das eingebettete Expertenwissen und proklamiert, dass ein qualitativer Mehrwert erreicht und mittels einer geeigneten Metrik gemessen werden kann. Die Qualität einer Notrufabwicklung spiegelt sich im Ergebnis wider, welches sowohl aus der Disposition der Einsatzmittel als auch in der Zeit zur Abwicklung des Notrufs besteht. Da die Zeit mit einem vollkommen unbekanntem und neuem System, das ohne vorheriges Training mit den Teilnehmer:innen getestet wurde, keine Aussagekraft bietet, beschränkt sich die Evaluation von Hypothese 3 auf die Beurteilung des Dispositionsergebnisses. Wie in Kapitel 9.2.3 erläutert, wurde das optimale Ergebnis für jeden der vorbereiteten Fälle durch den ärztlichen Leiter der ILtS-Ludwigshafen festgelegt. Im Rahmen der Testdurchläufe konnten die Disponent:innen jeden der sieben möglichen Dispositionen nach eigenem Ermessen aktivieren oder darauf verzichten. Jede Übereinstimmung zwischen der Disposition und dem vorgegebenen optimalen Ergebnis wird mit einem Punkt bewertet. Somit spiegelt eine Punktzahl von 7 ein optimales Prozessergebnis wider.

Auf den folgenden Seiten werden für jeden der drei vorbereiteten Notfälle die Ergebnisse aller Teilnehmer:innen dargelegt. Jeder Notfall (A,B,C) wird dabei in einer eigenen Tabelle mit allen dazu durchgeführten 21 simulierten Notrufen zusammengefasst. Das Sortierkriterium ist dabei der Testdurchlauf des jeweiligen Falles. Im ersten Durchlauf wurde ausschließlich der bSNA-Ablauf durchgeführt, während im zweiten und dritten Durchlauf der eSNA-Ablauf, einschließlich der auf dem Expertenwissen aufbauenden KIS-Prozesselemente, zum Einsatz kam. Zu jedem Fall wird die Person (als ID), die Dispositionen von D1-D7 sowie die erreichte Punktzahl dokumentiert. Am Ende jeder Tabelle wird zudem das optimale Dispositionsergebnis als Referenz angegeben. Die Dispositionen von D1-D7 entsprechen: Hotline 116117, Krankentransport, Rettungstransport, Notarzt, Rettungshubschrauber, Feuerwehr, Handlungsempfehlungen. Im Anschluss an jede Tabelle wird das Ergebnis kurz erörtert.

Durchlauf	Person	D1	D2	D3	D4	D5	D6	D7	Punkte
1	1	-	-	x	x	-	-	-	5
1	2	-	-	x	x	-	-	-	5
1	7	-	-	x	x	-	-	-	5
1	8	-	-	x	x	-	-	-	5
1	13	-	-	x	x	-	-	-	5
1	14	-	-	x	x	-	-	-	5
1	19	-	-	x	x	-	-	-	5
1	20	-	-	x	x	-	-	-	5
2	3	-	-	x	x	x	-	x	7
2	5	-	-	x	x	x	-	x	7
2	9	-	-	x	x	x	-	x	7
2	11	-	-	x	x	x	-	x	7
2	15	-	-	x	x	x	-	x	7
2	17	-	-	x	x	x	-	x	7
2	21	-	-	x	x	x	-	x	7
3	4	-	-	x	x	x	-	x	7
3	6	-	-	x	x	x	-	x	7
3	10	-	-	x	x	-	-	-	5
3	12	-	-	x	x	x	-	x	7
3	16	-	-	x	x	x	-	-	6
3	18	-	-	x	x	x	-	x	7
Optimum		-	-	x	x	x	-	x	7

Tabelle 9.3: Dispositionsergebnisse für Fall A: Bewusstlosigkeit

Die Auswertung von Fall A zeigt sehr deutlich, wie die eSNA-Abläufe die Disponent:innen zu besseren Ergebnissen geleitet haben. Während im jeweils ersten Durchlauf nach bSNA die Anwender:innen die Bewusstlosigkeit nicht mit der Möglichkeit einer Gasvergiftung in Zusammenhang brachten, wurden mit der eSNA die möglichen Ursachen ins Blickfeld gerückt. In Folge der dadurch erweiterten Wissensakquisition wurde eine entsprechend angepasste Disposition durchgeführt. Durchlauf 1 erzielte im Durchschnitt 5 Punkte, während Durchlauf 2 und 3 auf einen durchschnittlichen Punktwert von 6,77 kommen.

Durchlauf	Person	D1	D2	D3	D4	D5	D6	D7	Punkte
1	3	-	-	x	x	-	-	-	4
1	4	-	-	x	x	-	-	-	4
1	9	-	-	x	-	-	-	-	5
1	10	-	-	x	-	-	-	-	5
1	15	-	-	x	-	-	-	-	5
1	16	-	-	x	x	-	-	-	4
1	21	-	-	x	x	-	-	-	4
2	1	-	-	x	-	-	-	-	5
2	6	-	-	x	-	-	-	-	5
2	7	-	-	x	-	-	-	-	5
2	12	-	-	x	-	-	-	-	5
2	13	-	-	x	-	-	-	-	5
2	18	-	-	x	-	-	-	-	5
2	19	x	-	-	-	-	-	-	7
3	2	-	-	x	-	-	-	-	5
3	5	x	-	-	-	-	-	-	7
3	8	x	-	-	-	-	-	-	7
3	11	-	-	x	x	-	-	-	4
3	14	x	-	-	-	-	-	-	7
3	17	-	-	x	-	-	-	-	5
3	20	x	-	-	-	-	-	-	7
Optimum		x	-	-	-	-	-	-	7

Tabelle 9.4: Dispositionsergebnisse für Fall B: Schmerzen Brustbereich

In Fall B wurde von den Anwender:innen eine wesentliche Reduktion der gewählten Maßnahmen erwartet. Die fiel vielen Teilnehmern schwer, denn der Verzicht auf eine Entsendung eines Rettungswagens kann im Zweifel ernste Konsequenzen haben und eine Überdisposition ist weniger angreifbar. In Folge dessen fällt das Punktergebnis weniger deutlich zu Gunsten der eSNA-Variante aus. Durchlauf 1 erzielte im Durchschnitt 4,43 Punkte, während Durchlauf 2 und 3 auf einen durchschnittlichen Punktwert von 5,64 kommen.

Durchlauf	Person	D1	D2	D3	D4	D5	D6	D7	Punkte
1	5	-	-	x	x	-	-	-	6
1	6	-	-	x	x	-	-	-	6
1	11	-	-	x	x	-	-	-	6
1	12	-	-	x	x	-	-	-	6
1	17	-	-	x	-	-	-	-	5
1	18	-	-	x	-	-	-	-	5
1	2	-	-	x	-	-	-	-	5
1	4	-	-	x	-	-	-	-	5
2	8	-	-	x	x	-	-	-	6
2	10	-	-	x	x	-	-	-	6
2	14	-	-	x	x	-	-	-	6
2	16	-	-	x	x	-	-	-	6
2	20	-	-	x	x	-	-	-	6
2	1	-	-	x	-	-	-	-	5
2	3	-	-	x	-	-	-	-	5
3	7	-	-	x	x	x	-	-	7
3	9	-	-	x	x	x	-	-	7
3	13	-	-	x	-	-	-	-	5
3	15	-	-	x	x	-	-	-	6
3	19	-	-	x	x	-	-	-	6
3	21	-	-	x	x	-	-	-	6
Optimum		-	-	x	x	x	-	-	7

Tabelle 9.5: Dispositionsergebnisse für Fall C: Verkehrsunfall

In Fall C wurde den Disponent:innen eine zunächst harmlos klingende Situation geschildert. Im weiteren Verlauf blieben jedoch auch innerhalb des ersten Durchlaufs nach bSNA Unklarheiten, worauf sicherheitshalber mit einer dem Optimum nahen Disposition reagiert wurde. Im Rahmen der Durchläufe nach eSNA wurde der Schweregrad der Situation deutlicher erkennbar, was jedoch lediglich die Sicherheit hinsichtlich einer vergleichbaren Dispositionen erhöhte, nicht das Ergebnis selbst. Die Annahme, dass die harmlos klingende Beschreibung zum Beginn des Notrufs zu einer Unterdisposition führt, hat sich nicht bestätigt. Entsprechend sind auch kaum nennenswerte Punktunterschiede zu erkennen. Durchlauf 1 erzielte im Durchschnitt 5,7 Punkte, während Durchlauf 2 und 3 auf einen durchschnittlichen Punktwert von 5,8 kommen.

Zusammenfassend kann festgestellt werden, dass die Verbesserung der Dispositionsqualität je nach simuliertem Notfall sehr stark variiert.

- A - die Qualität konnte von 71,4% um 25,3% auf 96,7% gesteigert werden.
- B - die Qualität konnte von 63,3% um 17,3% auf 80,6% gesteigert werden.
- C - die Qualität konnte von 81% um 1,9% auf 82,9% gesteigert werden.

Die Hypothese 3 kann aufgrund dieser Daten dennoch als bestätigt angesehen werden, da nachweislich unter Ausnutzung des eingebetteten Expertenwissens eine teilweise signifikante qualitative Steigerung des Prozessergebnisses im Vergleich zu dem etablierten Verfahren gemessen werden konnte.

Die Einordnung der Evaluation in den Kontext der durch diese Arbeit adressierten Forschungsfragen wird im nachfolgenden Kapitel als Fazit vorgenommen.

10 Fazit

In diesem abschließenden Kapitel werden die Ergebnisse der Dissertation zusammengefasst und präsentiert. In Abschnitt 10.1 wird hierzu nochmals der Kontext skizziert, in dessen Rahmen diese Arbeit durchgeführt wurde. Dabei werden, ausgehend von der Forschungslücke hinsichtlich wissensintensiver Prozesse, der Ansatz dieser Arbeit und die entwickelten Artefakte kurz rekapituliert. Mit Abschnitt 10.2 wird der wissenschaftliche Beitrag der Dissertation zusammenfassend dargestellt. Anschließend werden die in der Einleitung eingeführten Forschungsfragen in Abschnitt 10.3 nochmals aufgegriffen und mit den Ergebnissen aus Kapitel 9 beantwortet. Hierbei wird auch die Übertragbarkeit der Ergebnisse auf wissensintensive Prozesse anderer Domänen diskutiert. In 10.4 werden dann zukünftige und an diese Arbeit anknüpfende Forschungsfragen identifiziert, um die bestehenden Forschungslücken weiter schließen zu können. Im abschließenden Abschnitt 10.5 wird dann der Beitrag dieser Arbeit mit der gesellschaftlichen Relevanz des untersuchten Anwendungsfalls verknüpft und mit Perspektiven zur Fortführung verbunden.

10.1 Zusammenfassung

Mit Blick auf den fortschreitenden gesellschaftlichen Wandel [51][82, S. 1-6] ist es zu erwarten, dass bestehende und sich noch entwickelnde wissensintensive Prozesse [43] zunehmend an Bedeutung gewinnen. Die aktive Forschung mit dem Schwerpunkt zu KiPs [114, 113, 8, 47, 28, 70, 60] zeigt zudem auf, dass noch nach geeigneten und ausbaufähigen Ansätzen gesucht wird, um dieser besonderen Art von Geschäftsprozessen gerecht werden zu können. Auf Grundlage dieser und weiterer Arbeiten sind die damit einhergehenden, besonderen Charakteristiken und Anforderungen identifiziert [32] und eine Reihe bestehender und teils etablierter Ansätze sind hinsichtlich ihrer Eignung für KiPs untersucht worden. Dabei zeigte sich deutlich eine noch bestehende Forschungslücke, welche im Rahmen dieser Arbeit adressiert wurde.

Hierzu wurden vor allem drei zentrale Anforderungen an KiPs fokussiert: Flexibilität, Adaptivität und Zielorientierung. Innerhalb des BPM-Forschungsbereichs wurden entsprechend dieser Anforderungen drei besonders geeignet erscheinende Grundprinzipien identifiziert. Mit einem deklarativen Prozessmodellierungsansatz geht dabei die Erwartung einher, dass eine flexible Prozessausführung erreicht werden kann. Von der Umsetzung eines datenorientierten Ansatzes wird erwartet, dass sich eine Prozessausführung an dem jeweiligen Kenntnisstand zu einem Prozess ausrichtet. Mit Hilfe einer semantischen Prozessmodellierung soll zudem der besonderen Rolle von Daten, Information und Wissen im Kontext wissensintensiver Prozesse Rechnung getragen werden. Die grundlegende Strategie dieser Arbeit war es dabei, diese drei Grundprinzipien in einem neuen, einheitlichen Ansatz namens ODD-BP zu kombinieren und dadurch die jeweiligen Vorteile zu vereinen.

Der designorientierten Forschungsmethodik folgend wurden hierzu drei zentrale Artefakte konzipiert. Mit dem Metamodell (Artefakt 1) aus Kapitel 5 wurde der angestrebte datenorientierte und deklarative Charakter des ODD-BP Ansatzes geprägt. Mit der Methodik zur Inferenzierung von Prozesszuständen (Artefakt 2) aus Kapitel 6 wurde zudem ein Verfahren entwickelt, welches sich zu jedem Zeitpunkt an eine durch die Daten repräsentierte neue Situation anpassen kann und somit eine flexible sowie zielorientierte Prozessausführung bietet. Darüber hinaus können durch eine adaptive Prozessdarstellung (Artefakt 3) aus Kapitel 7 vielversprechende Ausführungsoptionen empfohlen werden, um die Anwender:innen während der Prozessausführung zu unterstützen. Letztlich wurden anschließend in Kapitel 8 (Artefakt 4) die ersten drei Artefakte zusammengefasst und in einem prototypischen System realisiert. Auf Basis dieser Artefakte wurde der ODD-BP Ansatz umgesetzt und im Kontext eines wissensintensiven Anwendungsfalls einer Notrufabfrage innerhalb der Domäne der Leitstellen in Kapitel 9 evaluiert.

10.2 Wissenschaftlicher Beitrag

Der im Rahmen dieser Arbeit entwickelte ODD-BP Ansatz kombiniert die drei Grundprinzipien einer *datenorientierten, deklarativen, semantischen* Prozessmodellierung. Hierbei konnten die vorteilhaften Eigenschaften zur Ausführung wissensintensiver Prozesse hinsichtlich *Flexibilität, Adaptivität* und *Zielorientierung* in einem gemeinsamen einheitlichen Ansatz zusammengeführt werden. Durch die Ausrichtung auf einen spezifischen Anwendungsfall, der Notrufabfrage innerhalb von BOS-Leitstellen, und der Integration von Expertenwissen konnten zudem explizite Prozessbeiträge auf Basis symbolischer K.I. geleistet werden, was letztlich eine unmittelbare Arbeitsteilung zwischen Mensch und Maschine darstellt. Dabei sind die entwickelten Artefakte A1, A2, A3 und A4 grundsätzlich domänenunabhängig. Erst durch die Erweiterung um domänenspezifische Konzepte innerhalb der Domain-Ontologie sowie des durch SWRL-Regeln repräsentierten Expertenwissens, fand eine Ausrichtung auf das Anwendungsszenario im Leitstellenkontext statt. Durch den Austausch der Domain-Ontologie kann grundsätzlich jedes Anwendungsszenario in jeder beliebigen Domäne aufgegriffen und umgesetzt werden. Aufgrund der durch ODD-BP innerhalb des Metamodells repräsentierten Charakteristik eines deklarativen und datenorientierten Prozessmodells eignet sich der Ansatz besonders für wissensintensive Prozesse. Denkbar sind hierbei beispielsweise die in den Forschungsprojekten Semaflax und Semanas adressierten Einsatzbereiche des Baumängelmanagements innerhalb der Baubranche oder das Antragsverfahren von Fördermaßnahmen in der Landwirtschaft. Darüber hinaus finden sich zahlreiche wissensintensive Prozesse im Gesundheitssektor und weiteren Branchen.

Im Gegensatz dazu kann festgehalten werden, dass Prozesse mit einem überwiegend aktivitätzentrierten und strikten Ablauf von dem vorgestellten ODD-BP Ansatzes nicht profitieren und eine Umsetzung nach einem etablierten Ansatz wie BPMN vorteilhafter wäre. Dies beruht im Wesentlichen auf der Eigenschaft, dass ein strikter Prozessablauf in einem deklarativen Ansatz wie ODD-BP die Definition vieler Einschränkungen (engl. Constraints) erfordert und somit zu einem eher komplexen Prozessmodell führt, verglichen mit einem kontrollflussorientierten Ansatz.

Neben den bereits dargelegten Vorteilen des ODD-BP Ansatzes, welche direkt an den Anforderungen von KiPs ausgerichtet und durch die Forschungsfragen und Hypothesen adressiert wurden, sind noch weitere erwähnenswerte und vorteilhafte Aspekte sichtbar geworden.

Einheitliche Wissensbasis

Innerhalb etablierter PAIS und WfMS-Systeme wird Wissen aus unterschiedlichen Quellen verwendet, um eine Prozessunterstützung zu leisten. Betrachtet man den Programmcode einer Workflow-Engine, eine Datei zur Beschreibung eines Prozessmodells sowie die Verwaltung von Daten einer Prozessinstanz innerhalb einer Datenbank, so verteilt sich prozessrelevantes Wissen zumeist auf unterschiedliche Komponenten und Technologien, was einem Medienbruch gleich kommt. Dem gegenüber wurden beim ODD-BP Ansatz alle Komponenten mit prozessrelevantem Wissen (Metamodell, Domänenwissen sowie Wissen über Prozessdefinitionen und Prozessinstanzen) in eine *einheitliche Wissensbasis* integriert. Dies bietet die Möglichkeit, dass der Prozesszustand einer Prozessinstanz auf Grundlage dieser integrierten Wissensbasis inferenziert werden kann. Dadurch entfällt zum einen die Notwendigkeit zur Implementierung einer Workflow-Engine. Darüber hinaus wird zum anderen die Grundlage geschaffen, um mittels Expertenwissen einen unmittelbaren Prozesseinfluss ausüben zu können. Mit Hilfe einer umfassenden Menge modellierten Expertenwissens kann auf diesem Weg auch eine weitreichende Prozessunterstützung realisiert werden. Letztlich bietet zudem die semantische Prozessmodellierung durch eine strenge Formalisierung eine für Mensch und Maschine eindeutige Prozessbeschreibung, frei von Ambiguitäten.

Anpassbarkeit des Metamodells

Aufgrund der ebenfalls deklarativen Beschreibung des Metamodells innerhalb der Basis-Ontologie wird das zur Prozessmodellierung erforderliche Sprachkonstrukt definiert. Dies gestattet die Anpassung oder Erweiterung des Sprachkonstrukts, ohne dass hierzu Anpassungen an beispielsweise der Implementierung einer WF-Engine erforderlich werden. Auf diesem Weg können grundsätzlich auch andere Modellierungsparadigmen wie beispielsweise BPMN, EPC, Artifact ganz oder teilweise innerhalb des Sprachkonstrukts aufgenommen werden. Diese grundsätzliche Möglichkeit wurde hinsichtlich ihrer Umsetzbarkeit nicht im Rahmen dieser Arbeit weiter untersucht und motiviert zu daran anknüpfenden weiteren Forschungsarbeiten. Insbesondere die Möglichkeit einer Integrierbarkeit bestehender BPMN-Prozessmodelle und einer anschließenden sukzessiven Transformation zu einem deklarativen und datenorientierten Prozessmodell eröffnet eine interessante und vielversprechende Perspektive für Anwender:innen mit einem etablierten kontrollflussorientierten Ansatz.

Adaptivität des Prozessmodells

Auch wenn im Rahmen dieser Arbeit keine Adaptivität des Prozessmodells, sondern die Adaptivität der Prozessvisualisierung untersucht wurde, so eröffnet der ODD-BP Ansatz hierfür weitreichende Möglichkeiten. In der Regel wird die Anpassbarkeit eines Prozessmodells motiviert durch die Erwirkung von Flexibilität, so wie dies bei den in Kapitel 2.5.1 vorgestellten Verfahren der Fall ist. Aufgrund der inhärenten Flexibilität des ODD-BP Ansatzes ist dieses Motiv obsolet. Allerdings kann eine Adaption des Prozessmodells zur Anpassung oder Neuausrichtung von Prozesszielen von Interesse sein. Hierfür ist der vorgestellte Ansatz in besonderer Weise geeignet, insofern ein Prozessmodell repräsentiert wird durch eine Menge an Statements entsprechend der in Kapitel 2.7 eingeführten Subjekt-Prädikat-Objekt Formalisierung. Dadurch kann im einfachsten Fall eine Prozessinstanz um weitere Prozesselemente durch eine zusätzliche Menge an Statements erweitert werden. Hierbei ist auch eine eng verzahnte Integration in ein bestehendes Prozessmodell auf Basis einer semantischen Integration denkbar. Dieser Aspekt des ODD-BP Ansatzes wird im Rahmen zukünftiger Arbeiten weiter verfolgt werden.

10.3 Evaluationsergebnisse

Innerhalb der Evaluation wurden drei Hypothesen formuliert, welche die beiden zentralen Forschungsfragen aufgreifen und damit verbundene Erwartungen in einen evaluierbaren Rahmen überführen. Mit der ersten Forschungsfrage wurde eine grundlegende und domänenunabhängige Erwartungshaltung zum Ausdruck gebracht.

Forschungsfrage 1 (FF1)

Kann aus den drei dargelegten Grundprinzipien ein einheitlicher Ansatz entwickelt werden, der die Kernanforderungen wissensintensiver Prozesse hinsichtlich Flexibilität, Adaptivität und Zielorientierung erfüllt?

Entsprechend dem in Kapitel 3.4 eingeführten übergeordneten Rahmen dieser Arbeit stehen alle vier erarbeiteten Artefakte (A1-A4) in einem unmittelbaren Zusammenhang mit der ersten Forschungsfrage. A1, A2 und A3 definieren die grundlegenden Modelle, Methoden und Verfahren, welche mit A4 in einem prototypischen System realisiert wurden und dabei

die drei identifizierten Grundprinzipien einer deklarativen, datenorientierten und semantischen Prozessmodellierung in einem einheitlichen Ansatz vereint haben. Im Rahmen von FF1 wird dabei die Frage formuliert, ob diese Kombination in einem einheitlichen Ansatz dazu dient, die Kernanforderungen wissensintensiver Prozesse hinsichtlich *Flexibilität*, *Adaptivität* und *Zielorientierung* zu erfüllen.

Mit der ersten Hypothese (Kapitel 9.3) wird dabei die Anforderung hinsichtlich einer *flexiblen Prozessausführung* aufgegriffen und mit dem prototypischen System in dem eingeführten Anwendungsszenario einer Notrufabfrage evaluiert. Die Ergebnisse zeigen deutlich, dass der ODD-BP Ansatz im Vergleich zum State-of-the-Art Ansatz eine flexible Prozessausführung unterstützt. Dies ist zumindest gültig für das untersuchte Anwendungsszenario. Die Übertragbarkeit auf weitere Anwendungsfälle und Domänen kann grundsätzlich angenommen werden, sollte allerdings mit zukünftigen Forschungsarbeiten weiter untersucht und adressiert werden.

Die zweite Hypothese (Kapitel 9.4) fokussiert die Kernanforderungen hinsichtlich *Adaptivität* und *Zielorientierung* und der damit verbundenen Erwartung, dass die Anwender:innen dies als eine Unterstützung bei der Prozessausführung einstufen. Zur Bewertung dieser Hypothese wurde eine empirische Evaluation in Form einer Anwenderbefragung durchgeführt, im Rahmen dessen einzelne Fragen gezielt die genannten Aspekte adressierten. Die positiven Ergebnisse liefern zwar erwartungsgemäß keinen eindeutigen Beweis zur zweiten Hypothese, erlauben es aber, diese als sehr plausibel einzustufen. Mit den Fähigkeiten zur Adaptivität und Zielorientierung geht die Möglichkeit einher, dass wesentlich komplexere Prozessabläufe realisiert und für die mitwirkenden Personen beherrschbar werden. Ein Aspekt, der im Rahmen dieser Arbeit nicht weiter vertieft werden konnte und mit weiteren Forschungsarbeiten aufgegriffen werden sollte.

Die Ergebnisse beider Hypothesen liefern somit einen Beitrag hin zu einer Bestätigung der ersten Forschungsfrage. Ja, die drei dargelegten Grundprinzipien können in einem Ansatz vereint werden und erfüllen dabei die Kernanforderungen wissensintensiver Prozesse im untersuchten Anwendungsszenario. Eine Übertragbarkeit auf andere Anwendungsfälle und Domänen ist naheliegend und sollte durch zukünftige Arbeiten untersucht werden.

Mit der zweiten Forschungsfrage wird für den neuen Ansatz ein domänenspezifischer Bezug hergestellt.

Forschungsfrage 2 (FF2)

Kann in dem einheitlichen Ansatz ontologisches Wissen so integriert werden, dass dieses in einer Prozessausführung einen Beitrag leistet?

Im übergeordneten Rahmen (Kapitel 3.4) wird der Zusammenhang zwischen den Artefakten A2 und A4 zur zweiten Forschungsfrage hergestellt. Hierzu wird in das entwickelte prototypische System (A4) Expertenwissen eingebracht, welches bei der Ausführung der Inferenzierungsmethodik (A2) berücksichtigt wird. Durch das integrierte Expertenwissen, welches ausgerichtet ist auf das Anwendungsszenario, wird ein direkter Domänenbezug innerhalb der Leitstelle hergestellt.

Mit der dritten Hypothese (Kapitel 9.5) wird FF2 aufgegriffen und es wird proklamiert, dass durch das eingebettete Expertenwissen ein qualitativer Mehrwert erreicht werden kann. Die Evaluation basiert zum einen auf einer auf das Anwendungsszenario maßgeschneiderten Metrik, um die Qualität des Prozessergebnisses mit einem Zahlenwert ausdrücken zu können. Zum anderen wurden die Prozessergebnisse nach dem etablierten Basisverfahren (bSNA) und dem für die Evaluation entwickelten erweiterten Verfahren (eSNA) erfasst. Zur Evaluation wurde dabei mit drei simulierten Notfällen gearbeitet und die Verbesserung der Dispositionsergebnisse zwischen bSNA und eSNA wurde für jeden Fall einzeln miteinander verglichen.

Auch wenn die Verbesserung der Dispositionsergebnisse je nach Notfall variiert, so zeigen die Ergebnisse klar, dass ontologisches Wissen teilweise einen signifikanten Beitrag bei einer Prozessausführung leisten kann. Allerdings erlauben sowohl die Metrik als auch das Expertenwissen und der Anwendungsfall nur eine Aussage über die simulierten Notfälle. Die zugrundeliegende Methodik erlaubt zwar grundsätzlich eine Übertragbarkeit auf andere Anwendungsfälle und Domänen, allerdings werden die messbaren Ergebnisse dominiert durch die jeweilige Umsetzung eines Anwendungsfalls und des damit eingebundenen Expertenwissens. Interessant wäre eine längerfristige Untersuchung einer Qualitätssteigerung in einer Domäne, in welcher jeweils zwei im Vergleich stehende Prozesse parallel ausgeführt werden könnten.

10.4 Zukünftige Arbeiten

Die Evaluationsergebnisse zeigen, dass der ODD-BP Ansatz vielversprechende Eigenschaften bietet, um wissensintensive Prozesse auszuführen und dabei eine flexible, adaptive und zielorientierte Prozessausführung zu gestatten. Eine Übertragbarkeit auf andere Domänen und Anwendungsfälle ist naheliegend, sollte jedoch mit weiteren Studien untersucht und belegt werden. Hierbei wäre es zudem interessant, eine Metrik zu entwickeln, um Prozessmodelle hinsichtlich ihrer generellen Eignung für einen kontrollflussorientierten oder einen datenorientierten Ansatz einzustufen zu können. Nachfolgend werden in einzelnen Absätzen zudem mehrere Anknüpfungspunkte identifiziert, um die dieser Arbeit zugrundeliegenden Forschungslücken zu adressieren, die Antworten zu den formulierten Forschungsfragen zu vervollständigen und den neu entwickelten ODD-BP Ansatz weiterzuentwickeln.

Im Kontext des untersuchten Anwendungsfalls kann beobachtet werden, dass ein Notruf ohne Kenntnis über das eigentliche Prozessziel startet. Klassische Verfahren stoßen hier an Grenzen, da in der Regel mit der Auswahl eines Prozessmodells auch die Auswahl eines Prozessziels einhergeht. Adaptive und zielorientierte Ansätze wie ODD-BP versprechen einen deutlichen Vorteil zu bieten und zugleich die Möglichkeit zu eröffnen, wesentlich komplexere Prozessmodelle ausführen und beherrschen zu können. Mit zukünftigen Arbeiten sollten Prozesse identifiziert werden, die ohne Kenntnis über das eigentliche Prozessziel starten. Hierbei könnte die Eignung des ODD-BP Ansatzes für solche Prozesse sowie die generelle Übertragbarkeit auf unterschiedliche Domänen weiter untersucht werden.

In dieser Arbeit wurde das Expertenwissen in Form von SWRL-Regeln auf konzeptioneller Ebene in die einheitliche Wissensbasis eingebracht. Dies ist mit einer Komplexität verbunden, welche einer einfachen und umfassenden Integration solchen Wissens entgegen steht. Anknüpfend an die bisherigen Ergebnisse sollte im Rahmen zukünftiger Arbeiten ein Regelwerk erarbeitet werden, wie Expertenwissen in einer einfachen Form in ein System eingebracht und zur Laufzeit in ausführbare Regeln überführt werden kann. Die bisherigen Ergebnisse legen nahe, dass mit einer umfangreichen Menge an entsprechenden Regeln auch eine weitreichende KI-gestützte Prozessunterstützung möglich ist, was durch weitere Forschungsarbeiten belegt werden sollte.

Innerhalb von Kapitel 7 wurden Verfahren entwickelt, damit die durch ODD-BP gebotene Flexibilität nicht zu einem weitreichenden Anstieg an Komplexität und zu einem Verlust an Prozesssupport führt. Auch Adaptionen innerhalb der Prozessdarstellung können eine Herausforderung für Anwender:innen darstellen, sobald diese nicht zu einer etablierten Mentalmap eines Prozesses passen. Im Zuge zukünftiger Forschungsarbeiten sollten deshalb auch die Auswirkungen von Adaptionsvorgängen untersucht werden, um bei einer gebotenen, weitreichenden Flexibilität durch geeignete Prozessempfehlungen den nutzenden Personen eine gute Prozessunterstützung bieten zu können.

Mit der vorgestellten Möglichkeit zur Adaption eines Prozessmodells durch das Hinzufügen einer Menge zusätzlicher Subject-Prädikat-Object Statements können sowohl einfache als auch weitreichende Prozessmodifikationen ermöglicht werden. Eine Herausforderung besteht dabei in der semantischen Integration der hinzugefügten Prozesselemente mit den Elementen einer bestehenden und teilweise bereits ausgeführten Prozessinstanz. Es ist geplant, im Rahmen zukünftiger Arbeiten diese Herausforderung einer semantischen Integration aufzugreifen, um die Adaptivität, welche sich bislang auf die Prozessdarstellung begrenzt, auf das Prozessmodell erweitern zu können.

Im Zuge dieser Arbeit wurden die Herausforderungen bei Schlussfolgerungen unter der Open-World-Assumption deutlich. Die Möglichkeit eines gezielten *Closures* innerhalb einzelner Sachverhalte steht dabei in einem engen Zusammenhang mit der Komplexität eines abgebildeten Prozessmodells. Mit zukünftigen Arbeiten sollte ein Ansatz entwickelt werden, um die Inferenzierbarkeit durch Closures zu erhöhen, ohne dass damit eine gesteigerte Komplexität für die Prozessdesigner:innen einhergeht.

Letztlich wurde der im Rahmen dieser Arbeit vorgestellte ODD-BP Ansatz ausgerichtet auf die in den Forschungsfragen adressierten Herausforderungen konzipiert. Mit Blick auf die in Kapitel 2.4 vorgestellten etablierten Prozessmodellierungsansätze sollte im Rahmen zukünftiger Forschungsarbeiten die Ausdrucksmächtigkeit und Vollständigkeit weitergehend untersucht werden.

10.5 Perspektiven

Im Zuge der Evaluation in Zusammenarbeit mit der ILtS-Ludwigshafen hat sich innerhalb der Leitstellen bereits der klare Bedarf an einer umfassenden Unterstützung der Disponenten im Zusammenhang mit der Notrufbearbeitung gezeigt. Parallel dazu verdeutlichen Studien [126, 90] einen weitreichenden Bedarf und gleichzeitig verschiedene Ansatzpunkte, um mit Methoden der Künstlichen Intelligenz unterschiedliche Abläufe im Leitstellenkontext zu verbessern. Vor diesem Hintergrund und mit Blick auf aktuelle Ereignisse wie einer andauernden Pandemie, Hochwasserkatastrophen oder großflächigen Waldbrandereignissen wird der gesellschaftliche Wert des Katastrophen- und Rettungsschutzes und damit die Bedeutung einer gut funktionierenden Koordination durch die BOS-Leitstellen nochmals besonders deutlich. Dabei stellt die im Rahmen dieser Arbeit fokussierte Notrufabfrage in der Regel den ersten Kontakt zwischen den Betroffenen und notwendigen Hilfsmaßnahmen dar. Eine bestmögliche Bearbeitung solcher Notrufe liefert die Grundlage für eine schnelle und zielgerichtete Auswahl von Maßnahmen und kann Leiden und Schäden mindern und in Einzelfällen Leben retten.

Dabei können Notrufe nicht immer als ein begrenzter Einzelfall betrachtet werden. Gerade bei großen (z.B. Brand) oder weitreichenden (z.B. Pandemie) Ereignissen müssen Notrufe in einem entsprechend größeren Rahmen betrachtet und bearbeitet werden können. In diesem Kontext wurde ein Forschungsprojekt namens SPELL - Semantische Plattform zur intelligenten Entscheidungs- und Einsatzunterstützung in Leitstellen und beim Lagemanagement - beantragt und nach dessen Bewilligung im Juni 2021 gestartet. Im Rahmen dieses Projektes wird die hier vorgestellte Arbeit fortgeführt und hinsichtlich ihres potentiellen Nutzens nochmals wesentlich erweitert. Mit der Perspektive, dass ein umfangreicher Datenbestand in Form einer semantischen Plattform zur Verfügung steht, kann eine intelligente Notrufabfrage den Disponenten einer Leitstelle eine substantielle Unterstützung anbieten. Hierzu wird in Kooperation mit den Experten der Leitstellen ein umfassender Bestand an Expertenwissen aufgebaut, welcher auf Grundlage der in dieser Arbeit entwickelten Methoden und Verfahren Wirkung entfalten soll. Ziel ist es, mit Hilfe eines KI-gestützten Mehrwertdienstes eine fundierte und umfassende Verbesserung bei der Bearbeitung einzelner Notrufe bis hin zur Bewältigung weitreichender Ereignisse zu erwirken.

Schlusswort

Die im Rahmen dieser Arbeit entwickelten Methoden und Verfahren bieten unter dem Bezeichner ODD-BP einen vielversprechenden Ansatz, um wissensintensive Prozesse unterschiedlicher Domänen modellieren, visualisieren und ausführen zu können. Der Einsatz Semantischer Technologien bildet dabei das zentrale Fundament und unterstützt eine eindeutige Prozessbeschreibung, sowohl für die involvierten Personen als auch für eine maschinelle Verarbeitung. Hierzu kann auf Grundlage verfügbaren und innerhalb von Prozessen nutzbaren Expertenwissens ein substantieller Prozessbeitrag durch Methoden der Künstlichen Intelligenz geleistet werden. Im Rahmen des SPELL Forschungsprojektes wird der Ansatz dieser wissenschaftlichen Arbeit weiterentwickelt, um zeitnah einen praktischen Nutzen in einem gesellschaftlich höchst relevanten Bereich erwirken zu können.

Literaturverzeichnis

- [1] W M P Van Der Aalst and M Pesic. DecSerFlow : Towards a Truly Declarative Service Flow Language. 2006.
- [2] Sunitha Abburu. A Survey on Ontology Reasoners and Comparison. *International Journal of Computer Applications*, 57(17):33–39, 2012. doi:10.5120/9208-3748.
- [3] Ruth Sara Aguilar-Savén. Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2):129–149, 2004. doi:10.1016/S0925-5273(03)00102-6.
- [4] Benjamin Albrecht, Philip Effinger, Markus Held, and Michael Kaufmann. An automatic layout algorithm for BPEL processes. In *Proceedings of the 5th international symposium on Software visualization*, pages 173–182, 2010.
- [5] Thomas Allweyer. *BPMN 2.0 - Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung*. 2015.
- [6] Steven Alter. *Information Systems: A Management Perspective*. Addison-Wesley Reading, MA, 1999.
- [7] Autorengruppe. Formalisierung. In *dtv-Lexikon*. 1992.
- [8] T. Basten, M. Voorhoeve, and J. Filipe. Adaptive workflow: On the interplay between flexibility and support. *Enterprise Information Systems*, pages 63–70, 2000.
- [9] Sean Bechhofer. OWL Reasoning Examples, 2020. URL: <http://owl.man.ac.uk/2003/why/latest/>.
- [10] Fabian Beck, Michael Burch, and Stephan Diehl. Towards an aesthetic dimensions framework for dynamic graph visualisations. *Proceedings of the International Conference on Information Visualisation*, pages 592–597, 2009. doi:10.1109/IV.2009.42.

- [11] J Becker and R Schütte. *Handelsinformationssysteme*. Redline Wirtschaft bei moderne industrie. mi-Wirtschaftsbuch, 2004. URL: <https://books.google.de/books?id=LbXrCQ-aIcUC>.
- [12] Jörg Becker. Geschäftsprozessmodellierung (accessed: 26.05.2021). In *Enzyklopädie der Wirtschaftsinformatik*. Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik., 2021. URL: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Geschäftsprozessmodellierung/index.html>.
- [13] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 21, 2001. doi:10.1007/978-1-4020-4749-5_3.
- [14] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):7, 2002. arXiv:1204.6441, doi:10.1038/scientificamerican0501-34.
- [15] Stefanie Betz, Daniel Eichhorn, Susan Hickl, Stefan Klink, Agnes Koschmider, Yu Li, Andreas Oberweis, and Ralf Trunko. 3D representation of business process models. *Modellierung betrieblicher Informationssysteme (MobIS 2008)*, pages 73–87, 2008.
- [16] Stefanie Betz, Stefan Klink, Agnes Koschmider, and Andreas Oberweis. Automatic User Support for Business Process Modeling. In *Proceedings of the Workshop on Semantics For Business Process Management*, pages 1–12. 2006.
- [17] Kamal Bhattacharya, Cagdas Gerege, Richard Hull, Rong Liu, and Jianwen Su. Towards formal analysis of artifact-centric business process models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4714 LNCS:288–304, 2007.
- [18] Kamal Bhattacharya, Richard Hull, and Jianwen Su. A data-centric design methodology for business processes. *Handbook of Research on Business Process Modeling*, pages 503–531, 2009.
- [19] R Bobrik, T Bauer, and Manfred Reichert. Proviado - Personalized and Configurable Visualizations of Business Processes. *E-Commerce and Web Technologies*, 4082:61–71, 2006. doi:10.1007/11823865.

- [20] Ralph Bobrik. *Konfigurierbare Visualisierung komplexer Prozessmodelle*. PhD thesis, 2008. URL: <http://dbis.eprints.uni-ulm.de/438/>.
- [21] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. View-based process visualization. *Business Process Management*, pages 88–95, 2007. URL: <http://www.springerlink.com/index/q32382tj17k01408.pdf>, doi:10.1007/978-3-540-75183-0_7.
- [22] Conrad Bock, A. Fokoue, Peter Haase, R. Hoekstra, Ian Horrocks, Alan Ruttenberg, Uli Sattler, and Michael Smith. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition): W3C Recommendation, 2012. URL: <https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/>.
- [23] BPTrends. Case Management : Combining Knowledge With Process. *BPTrends*, (July):1–14, 2009. URL: <https://www.bptrends.com/bpt/wp-content/publicationfiles/07-09-WP-CaseMgt-CombiningKnowledgeProcess-White.doc-final.pdf>.
- [24] Ross Brown, Jan Recker, and Stephen West. Using virtual worlds for collaborative business process modeling. *Business Process Management Journal*, 17(3):546–564, 2011. doi:10.1108/14637151111136414.
- [25] Brenda J Burkhart and Marc E Fusco. Using animation to aid process flow visualization. In *Conference companion on Human factors in computing systems*, pages 21–22, 1996.
- [26] Manuel Castells. *Jahrtausendwende: Das Informationszeitalter (Band 3)*. 2004.
- [27] Volker Claus and Andreas Schwill. *Duden - Informatik*. Dudenverlag Mannheim, 1989.
- [28] David Cohn and Richard Hull. Business artifacts: A data-centric approach to modeling business operations and processes. *IEEE Data Eng. Bull.*, 32(3):3–9, 2009. URL: <http://scholar.google.com/scholar?hl=en%257B%257DbtnG=Search%257B%257Dq=intitle:Business+Artifacts+:+A+Data-centric+Approach+to+Modeling+Business+Operations+and+Processes%257B#%257D0>, doi:10.1007/978-3-540-88873-4_17.

- [29] Thomas H. Davenport. *Thinking for a Living: How to Get Better Performances And Results from Knowledge Workers*. Harvard Business School Press, Boston, 2005.
- [30] Andreas Dengel. *Semantische Technologien: Grundlagen - Konzepte - Anwendungen*. Spektrum Akademischer Verlag, 2012.
- [31] Claudio Di Ciccio, Andrea Marrella, and Alessandro Russo. Knowledge-intensive Processes: An overview of contemporary approaches. *CEUR Workshop Proceedings*, 861:33–47, 2012.
- [32] Claudio Di Ciccio, Andrea Marrella, and Alessandro Russo. Knowledge-Intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. *Journal on Data Semantics*, 4(1):29–57, 2015. doi:10.1007/s13740-014-0038-4.
- [33] Marlon Dumas, Wil M.P. van der Aalst, and Arthur H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. 2005. doi:10.1002/0471741442.
- [34] Philip Effinger, Martin Siebenhaller, and Michael Kaufmann. An interactive layout tool for bpmn. In *2009 IEEE Conference on Commerce and Enterprise Computing*, pages 399–406. IEEE, 2009.
- [35] Daniel Eichhorn, Agnes Koschmider, Yu Li, Peter Stürzel, Andreas Oberweis, and Ralf Trunko. 3D support for business process simulation. *Proceedings - International Computer Software and Applications Conference*, 1:73–80, 2009. doi:10.1109/COMPSAC.2009.20.
- [36] Dirk Fahland, Daniel Lübke, Jan Mendling, Hajo Reijers, Barbara Weber, Matthias Weidlich, and Stefan Zugal. Declarative versus imperative process modeling languages: The issue of understandability. *Lecture Notes in Business Information Processing*, 29 LNBIP:353–366, 2009. doi:10.1007/978-3-642-01862-6_29.
- [37] Shaokun Fan, Zhimin Hua, Veda C. Storey, and J. Leon Zhao. A process ontology based approach to easing semantic ambiguity in business process modeling. *Data & Knowledge Engineering*, (102):57–77, 2016. doi:10.1016/j.datak.2016.01.001.
- [38] Peter Fettke. UML-basierte Modellierung (accessed: 09.06.2021). In *Enzyklopädie der Wirtschaftsinformatik*. Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik.

- URL: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaeten-der-Systementwicklung/Problemanalyse-/Objektorientierte-Modellierung/UML-basierte-Modellierung/index.html>.
- [39] Roland Gabriel. Informationssystem (accessed: 22.05.2021). In *Enzyklopädie der Wirtschaftsinformatik*. Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik. URL: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/uebergreifendes/Kontext-und-Grundlagen/Informationssystem/index.html/?searchterm=Informationssystem>.
- [40] Ankita Gaur and Vineet Richariya. A Layered Approach for Intrusion Detection Using Meta - modeling with Classification Techniques. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, 1(2):161–168, 2008.
- [41] Matthias Gottlieb. GoITSystems - Design Science, 2020. URL: <https://goitsystems.de/index.php/forschungsmethoden/design-science.html>.
- [42] Norbert Gronau, Claudia Müller, and Mathias Uslar. The KMDL Knowledge Management Approach: Integrating knowledge conversions and business process modeling. *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, 3336:1–10, 2004. doi:10.1007/978-3-540-30545-3_1.
- [43] Norbert Gronau and Edzard Weber. Management of Knowledge Intensive Business Processes. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3080:163–178, 2004. doi:10.1007/978-3-540-25970-1_11.
- [44] Thomas R Gruber. Technical Report KSL 92-71 Revised April 1993 A Translation Approach to Portable Ontology Specifications by A Translation Approach to Portable Ontology Specifications. *Knowledge Creation Diffusion Utilization*, 5(April):199–220, 1993. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.117.3273&rep=rep1&type=pdf>.
- [45] Lisa Grumbach, Eric Rietzke, Markus Schwinn, Ralph Bergmann, and Norbert Kuhn. SEMAFLEX-Semantic integration of flexible workflow

- and document management. In R. Krestel, D. Mottin, and E. Müller, editors, *CEUR Workshop Proceedings: Lernen, Wissen, Daten, Analysen*, volume 1670, pages 43–50, Potsdam, 2016.
- [46] Lisa Grumbach, Eric Rietzke, Markus Schwinn, Ralph Bergmann, and Norbert Kuhn. SEMANAS - Semantic Support for Grant Application Processes. *Lernen, Wissen, Daten, Analysen - LWDA*, (13):2017–2021, 2018.
- [47] Christian W. Günther, Manfred Reichert, and Wil M.P. P Van Der Aalst. Supporting flexible processes with Adaptive Workflow and Case Handling. *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, pages 229–234, 2008. doi:10.1109/WETICE.2008.15.
- [48] Hugo Haas and Allen Brown. Web Services Glossary (accessed: 10.06.2021). W3C Working Group. URL: <https://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.
- [49] Siegfried Häberle. *Das neue Lexikon der Betriebswirtschaftslehre*. 2008. doi:10.1524/9783486843682.
- [50] C. Hagen, D. Ratz, and R. Povalej. Towards Self-Organizing Knowledge Intensive Processes. *Journal of Universal Knowledge Management*, 0(2):148–169, 2005.
- [51] Michael Hammer and James Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*, volume 19. HarperCollins, updated and revised version 2001 edition, 1993. doi:10.2307/258943.
- [52] Bernd Heinrich, Marc Bewernik, Matthias Henneberger, Alexander Krammer, and Florian Lautenbacher. SEMPA – Ein Ansatz des Semantischen Prozessmanagements zur Planung von Prozessmodellen. *Wirtschaftsinformatik*, 50(6):445–460, 2008. URL: <https://epub.uni-regensburg.de/23173/1/heinrich.pdf>, doi:10.1007/s11576-008-0085-2.
- [53] Martin Hepp, Frank Leymann, John Domingue, Alexander Wahler, and Dieter Fensel. Semantic business process management: A vision towards using semantic web services for business process management. *Proceedings - ICEBE 2005: IEEE International Conference on e-Business Engineering*, (1), 2005. URL: <http://citeseerx.ist.psu.edu/>

- viewdoc/summary?doi=10.1.1.141.5322, doi:10.1109/ICEBE.2005.110.
- [54] Alan Hevner and Samir Chatterjee. Design Science Research in Information Systems. In *Design Research in Information Systems*, chapter 2, pages 9–21. Springer, 2010.
- [55] R Hevner Alan. A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, 19(2):87–92, 2007.
- [56] Seok Hee Hong, Peter Eades, Aaron Quigley, and Sang Ho Lee. Drawing algorithms for Series-Parallel digraphs in two and three dimensions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1547(971):198–209, 1999. doi:10.1007/3-540-37623-2_15.
- [57] Matthew Horridge. A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.3, 2011. URL: http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf.
- [58] Ian Horrocks, Peter F Patel-schneider, Harold Boley, Said Tabet, Benjamin Grosf, and Mike Dean. SWRL : A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member submission 21*, (May 2004):1–20, 2004.
- [59] Richard Hull, Elio Damaggio, Fabiana Fournier, Manmohan Gupta, Anil Nigam, Piyawadee Sukaviriya, and Roman Vaculin. Introducing the Guard-Stage-Milestone Approach for Specifying Business Entity Lifecycles. (257593):1–22, 2010.
- [60] Richard Hull, Anil Nigam, Piyawadee Noi Sukaviriya, Roman Vaculin, Elio Damaggio, Riccardo De Masellis, Fabiana Fournier, Manmohan Gupta, Fenno Terry Heath, Stacy Hobson, Mark Linehan, and Sridhar Maradugu. Business artifacts with guard-stage-milestone lifecycles. *Proceedings of the 5th ACM international conference on Distributed event-based system - DEBS '11*, (January):51, 2011. doi:10.1145/2002259.2002270.
- [61] Gregor Joeris. Decentralized and flexible workflow enactment based on task coordination agents. *Agent-oriented information systems 2000: proceedings*, page 41, 2000. URL: <http://books.google.com/books?hl=en&lr=&id=ByU0wmVIYq4C&>

oi=fnd&pg=PA41&dq=Decentralized+and+flexible+workflow+enactment+based+on+task+coordination+agents&ots=_puvepuPT1&sig=ugtNi8hG5aCnXXijKN-ef30ZsrI.

- [62] Jens Kolb. *Abstraction, Visualization, and Evolution of Process Models*. PhD thesis, Ulm, 2015.
- [63] Jens Kolb and Manfred Reichert. A Flexible Approach for Abstracting and Personalizing Large Business Process Models. *Applied Computing Review*, 13(1):6–17, 2013. URL: <http://dbis.eprints.uni-ulm.de/914/>.
- [64] Jens Kolb and Manfred Reichert. Data Flow Abstractions and Adaptations through Updatable Process Views. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing - SAC '13*, 2013. doi:10.1145/2480362.2480632.
- [65] Helmut Krcmar. Information (accessed: 26.02.2021). Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik. URL: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/daten-wissen/Informationsmanagement/Information->.
- [66] Simone Kriglstein and Stefanie Rinderle-Ma. Change visualizations in business processes: Requirements analysis. *GRAPP 2012 IVAPP 2012 - Proceedings of the International Conference on Computer Graphics Theory and Applications and International Conference on Information Visualization Theory and Applications*, (February):584–593, 2012. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84862214815&partnerID=40&md5=f3efa9b269a712325ae1cf9be08d26cd>, doi:10.5220/0003815505840593.
- [67] Vera Künzle, Barbara Weber, and Manfred Reichert. Object-aware business processes: Fundamental requirements and their support in existing approaches. *Frameworks for Developing Efficient Information Systems: Models, Theory, and Practice*, pages 1–29, 2013. doi:10.4018/978-1-4666-4161-7.ch001.
- [68] Carsten Maletzki, Eric Rietzke, Lisa Grumbach, Ralph Bergmann, and Norbert Kuhn. Utilizing Ontology-Based Reasoning to Support the Execution of Knowledge-Intensive Processes. In *BPM 2019 - AI4BPM*, page 12, 2019.

- [69] Mike Marin, Richard Hull, and Roman Vaculín. Data centric BPM and the emerging case management standard: A short survey. In *Lecture Notes in Business Information Processing*, volume 132 LNBIP, pages 24–30, 2013. doi:10.1007/978-3-642-36285-9-4.
- [70] Olivera Marjanovic and Ronald Freeze. Knowledge intensive business processes: Theoretical foundations and research challenges. *Proceedings of the Annual Hawaii International Conference on System Sciences*, (February 2011), 2011. doi:10.1109/HICSS.2011.271.
- [71] Andreas Meyer, Sergey Smirnov, and Mathias Weske. Data in Business Processes. *EMISA Forum*, 31(3):5–31, 2011.
- [72] Mirjam Minor, Ralph Bergmann, and Sebastian Görg. Adaptive Workflow Management in the Cloud -Towards a Novel Platform as a Service. *Proceedings of the ICCBR 2011 Workshops*, (974):131–138, 2011.
- [73] Debdoot Mukherjee, Pankaj Dhoolia, Saurabh Sinha, Aubrey J. Rembert, and Mangala Gowri Nanda. From informal process diagrams to formal process models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6336 LNCS:145–161, 2010. doi:10.1007/978-3-642-15618-2_12.
- [74] Dominic Müller, Manfred Reichert, and Joachim Herbst. Flexibility of Data-driven Process Structures. *Lecture Notes in Computer Science*, 4103:179–190, 2006. doi:10.1007/11837862.
- [75] Dominic Müller, Manfred Reichert, and Joachim Herbst. A new paradigm for the enactment and dynamic adaptation of data-driven process structures. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5074 LNCS:48–63, 2008. arXiv:9780201398298, doi:10.1007/978-3-540-69534-9_4.
- [76] Nicolas Mundbrod, Jens Kolb, and Manfred Reichert. Towards a system support of collaborative knowledge work. *Lecture Notes in Business Information Processing*, 132 LNBIP:31–42, 2013. doi:10.1007/978-3-642-36285-9_5.
- [77] Alessandro Teixeira Neto, Flávio Luiz de Silva Bussamra, and Henrique Araújo de Castro e Silva. A new metamodel for reinforced panels

- under compressive loads and its application to the fuselage conception. *Latin American Journal of Solids and Structures*, 11(2):223–244, 2014. doi:10.1590/S1679-78252014000200005.
- [78] Klaus North. *Wissensorientierte Unternehmensführung Wertschöpfung durch Wissen*, volume 3. 2003. arXiv:1011.1669, doi:10.1088/1751-8113/44/8/085201.
- [79] Markus Nüttgens. EPK (accessed: 10.06.2021). In *Enzyklopädie der Wirtschaftsinformatik*. Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik. URL: <https://www.encyklopaedie-der-wirtschaftsinformatik.de/wi-encyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaten-der-Systementwicklung/Problemanalyse-/Geschäftsprozessmodellierung/EPK/index.html>.
- [80] OMG. *BPMN 2.0.1 - Business Process Model and Notation*. 2.0.1 edition, 2013.
- [81] OMG. *BPMN 2.0.2 - Business Process Model and Notation*, 2013.
- [82] Hubert Österle. *Business Engineering. Prozeß- und Systementwicklung: Band 1: Entwurfstechniken*. Springer-Verlag Berlin Heidelberg, Wiesbaden, 1995.
- [83] Hubert Österle, Jörg Becker, Ulrich Frank, Thomas Hess, Dimitris Karagiannis, Helmut Krcmar, Peter Loos, Peter Mertens, Andreas Oberweis, and Elmar J Sinz. Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. *Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*, page 1, 2010.
- [84] Maja Pesic, Helen Schonenberg, and Wil M.P. van der Aalst. DECLARE: Full Support for Loosely-Structured Processes. *11th IEEE international enterprise distributed object computing conference (EDOC 2007)*, pages 287–287, 2007. doi:10.1109/edoc.2007.14.
- [85] Helen C. Purchase, Eve Hoggan, and Carsten Goerg. How Important Is the Mental Map? – An Empirical Investigation of a Dynamic Graph Layout Algorithm. *Graph Drawing*, pages 184–195, 2007. URL: http://link.springer.com/10.1007/978-3-540-70904-6_19, doi:10.1007/978-3-540-70904-6_19.
- [86] Klaus Reichenberger. *Kompendium semantische Netze : Konzepte, Technologie, Modellierung*. Springer, 2010.

-
- [87] Manfred Reichert. Enabling Flexible and Robust Business Process Automation for the Agile Enterprise. pages 203–220. URL: <http://dbis.eprints.uni-ulm.de/1593/1/reichert.pdf>.
- [88] Manfred Reichert, Jens Kolb, Ralph Bobrik, and Thomas Bauer. Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. *27th ACM Symposium On Applied Computing (SAC'12), 9th Enterprise Engineering Track (EE'12)*, pages 1653–1660, 2012. URL: <http://dbis.eprints.uni-ulm.de/773/>, doi:10.1145/2245276.2232043.
- [89] Manfred Reichert, Stefanie Rinderle, Ulrich Kreher, and Peter Dadam. Adaptive process management with ADEPT2. *Proceedings - International Conference on Data Engineering, (Icde)*:1113–1114, 2005. doi:10.1109/ICDE.2005.17.
- [90] Melanie Reuter-Oppermann, Felix Libner, and Volkmar Lang. Maschinelles Lernen und Künstliche Intelligenz in BOS-Leitstellen. Technical report, Fachverband Leitstellen e.V., 2020.
- [91] Eric Rietzke, Ralph Bergmann, and Norbert Kuhn. Adaptive business process visualization for a data and constraint-based workflow approach. In *CEUR Workshop Proceedings: LWDA*, volume 1917, 2017.
- [92] Eric Rietzke, Ralph Bergmann, and Norbert Kuhn. Semantically-Oriented Business Process Visualization for a Data and Constraint-Based Workflow Approach. In Ernest Teniente and Matthias Weidlich, editors, *Business Process Management Workshops*, volume 308 of *Lecture Notes in Business Information Processing*, pages 142–150, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-74030-0_10.
- [93] Eric Rietzke, Ralph Bergmann, and Norbert Kuhn. ODD-BP - an Ontology- and Data-Driven Business Process Model. In *Lernen, Wissen, Daten, Analysen - LWDA*, page 12, 2019.
- [94] Eric Rietzke, Carsten Maletzki, Ralph Bergmann, and Norbert Kuhn. Execution of Knowledge-Intensive Processes by Utilizing Ontology-Based Reasoning. *Journal on Data Semantics*, page 16, 2021. doi:10.1007/s13740-021-00127-w.

- [95] S B Rinderle, R Bobrik, M U Reichert, and T Bauer. Business Process Visualization - Use Cases, Challenges, Solutions. *Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS'06): Information System Analysis and Specification*, pages 204–211, 2006. URL: <http://doc.utwente.nl/66217/>.
- [96] Michael Rosemann and Michael Zur Muehlen. Evaluation of Workflow Management Systems - A Meta Model Approach. *Australasian Journal of Information Systems*, 6(1):1–20, 1998. doi:10.3127/ajis.v6i1.322.
- [97] Sebastian Schlauderer and Sven Overhage. BPMN (accessed: 09.06.2021). In *Enzyklopädie der Wirtschaftsinformatik*. Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik. URL: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Hauptaktivitaeten-der-Systementwicklung/Problemanalyse-Workflow-Modellierung/bpmn/bpmn/?searchterm=bpmn>.
- [98] Helen Schonenberg, Ronny Mans, Nick Russell, Nataliya Mulyar, and Wil van der Aalst. Process flexibility: A survey of contemporary approaches. In *Advances in Enterprise Engineering I*, pages 16–30. Springer, 2008.
- [99] Bastiaan Schönhage, Alex van Ballegooij, and Anton Elliëns. 3D gadgets for business process visualization—a case study. In *Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, pages 131–138, 2000.
- [100] David Schumm, Gregor Latuske, and Frank Leymann. State Propagation for Business Process Monitoring on Different Levels of Abstraction. *European Conference on Information Systems*, (Ecis):12, 2011. URL: <http://aisel.aisnet.org/ecis2011/18/>.
- [101] Axel Schwickert. Der Geschäftsprozeß als formaler Prozeß – Definition, Eigenschaften, Arten. *Reihe: Arbeitspapiere WI*, 0(1), 1996. URL: <http://wi.uni-giessen.de>.
- [102] Thomas Setzer and Martin Bichler. Orchestrierung von Web-Services (accessed: 10.06.2021). In *Enzyklopädie der Wirtschaftsinformatik*. Gronau, Norbert et al. (Hrsg.): Enzyklopädie der Wirtschaftsinformatik. URL: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/>

wi-enzyklopaedie/lexikon/is-management/Systementwicklung/
Softwarearchitektur/Middleware/Web-Service-Technologien/
Orchestrierung-von-Web-Services/index.html.

- [103] Alexander Streit, Binh Pham, and Ross Brown. Visualization Support for Managing Large Business Process Specifications. In Wil M P van der Aalst, Boualem Benatallah, Fabio Casati, and Francisco Curbera, editors, *Business Process Management: 3rd International Conference, BPM 2005, Nancy, France, September 5-8, 2005. Proceedings*, pages 205–219. Springer Berlin Heidelberg, 2005. URL: http://dx.doi.org/10.1007/11538394_14, doi:10.1007/11538394_14.
- [104] Rudi Studer, V. Richard Benjamins, and Dieter Fensel. Knowledge Engineering: Principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197, 1998. doi:10.1016/S0169-023X(97)00056-6.
- [105] Kozo Sugiyama. *Graph drawing and applications for software and knowledge engineers*, volume 11. World Scientific, 2002.
- [106] Oliver Thomas and Michael Fellmann. Semantic Process Modeling – Design and Implementation of an Ontology-based Representation of Business Processes. *Business & Information Systems Engineering*, 1(6):438–451, 2009. doi:10.1007/s11576-009-0201-y.
- [107] Oliver Thomas, Simon Hagen, Ulrich Frank, Jan Recker, Lauri Wessel, Friedemann Kammler, Novica Zarvic, and Ingo Timm. Global Crises and the Role of BISE. *Business and Information Systems Engineering*, 62(4):385–396, 2020. URL: <https://doi.org/10.1007/s12599-020-00657-w>, doi:10.1007/s12599-020-00657-w.
- [108] Efraim Turban, Jay Aronson, E, and Ting-Peng Liang. *Decision Support Systems and Intelligent Systems*. 2007.
- [109] Mike Uschold and Michael Gruninger. ONTOLOGIES: Principles, Methods and Applications. *Knowledge Engineering Review*, (February), 1996.
- [110] Roman Vaculin, Richard Hull, Terry Heath, Craig Cochran, Anil Nigam, and Piyawadee Sukaviriya. Declarative business artifact centric modeling of decision and knowledge intensive business processes. In *15th IEEE International EDOC Conference Workshops*, pages 151–160. IEEE Computer Society, Los Alamitos, California, 2011. doi:10.1109/EDOC.2011.36.

- [111] W. M P Van Der Aalst, M. Pesic, and H. Schonenberg. Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23(2):99–113, 2009. doi: 10.1007/s00450-009-0057-9.
- [112] W. M P Van Der Aalst and A. H M Ter Hofstede. YAWL: Yet another workflow language. *Information Systems*, 30(4):245–275, 2005. doi: 10.1016/j.is.2004.02.002.
- [113] W. M.P. Van der Aalst and T. Basten. Inheritance of workflows: An approach to tackling problems related to change. *Theoretical Computer Science*, 270(1-2):125–203, 2002. doi:10.1016/S0304-3975(00)00321-2.
- [114] W. M.P. Van Der Aalst and S. Jablonski. Dealing with workflow change: Identification of issues and solutions. *Computer Systems Science and Engineering*, 15(5):267–276, 2000.
- [115] Wil M. P. van der Aalst. Process-Aware Information Systems: Design, Enactment, and Analysis. *Wiley Encyclopedia of Computer Science and Engineering*, pages 1–31, 2009. URL: <http://onlinelibrary.wiley.com/doi/10.1002/9780470050118.ecse577/full>, doi:10.1002/9780470050118.ecse577.
- [116] Wil M P van der Aalst. Business process management: A comprehensive survey. *ISRN Software Engineering*, 2013, 2013.
- [117] Wil M. P. Van Der Aalst, Mathias Weske, Dolf Grünbauer, Wil M P der Aalst, Mathias Weske, Dolf Grünbauer, Wil M. P. Van Der Aalst, and Mathias Weske. Case handling: A new paradigm for business process support. *Data and Knowledge Engineering*, 53(2):129–162, 2005. doi:10.1016/j.datak.2004.07.003.
- [118] R Hevner von Alan, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.
- [119] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview. *OWL 2 Web Ontology Language*, 2012.
- [120] Barbara Weber and Manfred Reichert. *Enabling Flexibility in Process-aware Information Systems*. 2012. doi:10.1007/978-3-642-30409-5.

-
- [121] Mathias Weske. *Business Process Management (Concepts, Languages, Architectures) 3rd Edition*. Springer, 2019. URL: https://www.google.de/books/edition/Business_Process_Management/-D5tpT5Xz8oC?hl=de&gbpv=1&dq=weske+business+process+management&printsec=frontcover.
- [122] Thomas G West, J Jerry Uhl, and James Martinez. Seeing the unseen: visualization in education. In *ACM SIGGRAPH 2006 Educators program*, pages 37—es. 2006.
- [123] Branimir Wetzstein, Zhilei Ma, Agata Filipowska, Monika Kaczmarek, Sami Bhiri, Silvestre Losada, Jose Manuel Lopez-Cobo, and Laurent Cicurel. Semantic Business Process Management: A lifecycle based requirements analysis. *CEUR Workshop Proceedings*, 251:1–11, 2007.
- [124] Wikipedia. Wikipedia: Formalisierung (accessed: 22.05.2021). In *Wikipedia.Formalisierung*. Wikipedia. URL: <https://de.wikipedia.org/wiki/Formalisierung>.
- [125] Stefan Zugal, Pnina Soffer, Cornelia Haisjackl, Jakob Pinggera, Manfred Reichert, and Barbara Weber. Investigating expressiveness and understandability of hierarchy in declarative business process models. *Software and Systems Modeling*, 14(3), 2015. arXiv:1511.04058, doi:10.1007/s10270-013-0356-2.
- [126] Zukunftsforum Öffentliche Sicherheit. Masterplan Leitstelle 2020, 2013. URL: <http://zoes-bund.de/wp-content/uploads/2015/10/Masterplan-Leitstelle-2020-ZOES.pdf>.