

**Ontologischer Ansatz zur abgestimmten
Anforderungserschließung für
Informationssysteme zur
Geschäftsprozessunterstützung**

Thomas Kofler

Institut für Informatik
der Technischen Universität München

**Ontologischer Ansatz zur abgestimmten
Anforderungserschließung für
Informationssysteme zur
Geschäftsprozessunterstützung**

Thomas Kofler

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen
Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Florian Matthes

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h.c. Manfred Broy
2. Univ.-Prof. Dr. Helmut Krcmar

Die Dissertation wurde am 09.01.2014 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik am 28.05.2014 angenommen.

Kurzfassung

Softwaresysteme werden traditionell in einem aus mehreren Phasen bestehendem Softwareentwicklungsprozess erstellt. Die erste Phase, das Requirements Engineering, ist eine problematische Phase. Präzise und vollständige Anforderungen sind schwer zu erheben und zusätzlich sind Fehler, die in dieser Phase gemacht werden, in späteren Phasen sehr teuer zu korrigieren.

Im Requirements Engineering sollen die Anforderungen an ein Softwaresystem so spezifiziert werden, dass sie so präzise, vollständig und konsistent wie möglich sind. Anforderungen basieren u.a. auf dem Wissen, das von den Gruppen der Nutzer, Entscheider und Fachexperten seitens des Auftraggebers durch Ermittlungstechniken erhoben wurde.

Anforderungsingenieure sind die Schnittstelle zwischen Auftraggeber und Auftragnehmer. Diese erfassen in frühen Phasen eine Vielzahl von Artefakten basierend auf dem Wissen der Stakeholder. Anforderungsingenieure sind für gewöhnlich selber keine Domänenexperten und können eventuell fehlendes Wissen in den Artefakten nicht durch die eigene Expertise ersetzen. Hinzu kommt, dass die Artefakte von den Anforderungsingenieuren häufig unabhängig voneinander erstellt werden und kein Gesamtverständnis über die Problem- und Lösungsdomäne existiert. Daraus resultiert eine Vielzahl von Problemen, die im Rahmen der Problemanalyse dieser Arbeit expliziert wird. U.a. befinden sich die erstellten Artefakte auf unterschiedlichen Abstraktionsebenen und sind deshalb nur schwer integrierbar und aufgrund fehlender Verknüpfung zwischen den Artefakten nicht untereinander auf ihre Konsistenz überprüfbar.

In dieser Arbeit wird eine Modellierungssprache entwickelt, die auf Inhalte von Artefakten basiert, die in frühen Phasen erstellt wurden. Aufbauend auf der Modellierungssprache wird eine Methodik erarbeitet, mit deren Hilfe ein Modell erstellt wird (das sog. Basismodell), das das gemeinsame Gesamtverständnis aller relevanten Stakeholder über die Problem- und Lösungsdomäne, bei der Entwicklung von prozessorientierten Informationssystemen, expliziert.

Aufbauend auf dem Basismodell zeigen wir beispielhaft wie Ausschnitte davon mit Hilfe anderer Modellierungstechniken detailliert werden können. Mit der Anwendung des Ansatzes erreichen wir u.a.: (a) eine Reduktion inhaltlicher Missverständnisse in frühen Phasen, (b) eine Reduktion von Inkonsistenzen und eine ausgeglichene Repräsentation von Wissen der mit dem Basismodell verknüpften Artefakte und (c) eine Repräsentation des stakeholderweiten Verständnisses.

Abschließend evaluieren wir den Ansatz in der Industrie sowie in der Anwendung in App-Entwicklungsprojekten im Rahmen von Master-Praktika an der TU München. Zur Absicherung der Ergebnisse führen wir zusätzlich ein Laborexperiment durch.

Abstract

Software systems are traditionally developed in a process consisting of several phases, where the first phase, requirements engineering, is a frequent source of problems. Accurate and complete requirements are difficult to obtain and errors made at this step are expensive to correct later on.

Requirement should be as accurate, complete and consistent as possible. Amongst others, these requirements are based on the knowledge of users, decision makers and the expertise of the client. Requirements engineers are at the interface between clients and contractors. Based on the knowledge of stakeholders, requirements engineers create a wide range of artifacts.

However, requirements engineers are typically no domain experts and thus unable to substitute potentially missing knowledge of artifacts. Additionally, artifacts are often created independent of other requirements engineers, therefore a comprehensive understanding of the problem and the solution domain does frequently not exist. This causes diverse problems which are discussed in this thesis. For example, artifacts may be on different levels of abstraction and due to a lack of linkage mutual consistency between artifacts cannot be ensured.

In this thesis a modeling language is developed which is based on artifacts created in early phases of requirements engineering. Based on this modeling language a methodology is developed that allows to create a model (the so called base model) which integrates the knowledge of all relevant stakeholders about the problem and solution domain in the development of process-oriented information systems.

We, furthermore, show exemplary how parts of the base model can be described in detail using other modeling techniques. By applying the approach suggested in this work we achieve i.a.: (a) a reduction of misunderstandings in early phases of requirements engineering (b) a reduction of inconsistencies and a more balanced representation of the knowledge captured in artifacts that are associated with the base model and (c) a representation of the stakeholder-wide understanding.

Finally, we evaluate this approach in the industry and in App development projects in the context of practical courses for master students at the TU Munich. We additionally made a laboratory experiment to consolidate the results presented in this work.

Danksagung

Besonders danke ich Prof. Dr. Dr. h.c. Manfred Broy für die Möglichkeit an seinem Lehrstuhl zu promovieren und für die Betreuung meiner Arbeit. Für mich war diese Zeit an der TU München einer der lehrreichsten Abschnitte meines Lebens. Vielen Dank dafür!

Herzlich danken möchte ich auch meinem Zweitgutachter, Prof. Dr. Helmut Krcmar, für die Bereitschaft, das Zweitgutachten meiner Arbeit zu übernehmen.

Auch möchte ich den vielen Kollegen danken, einerseits für die schönen Stunden und andererseits für die Unterstützung. Hervorheben möchte ich an dieser Stelle (in alphabetischer Reihenfolge): Georg Kalus, Marco Kuhrmann, Daniel Méndez Fernández, Silke Müller, Jakob Mund, Daniel Ratiu und Andreas Vogelsang.

Auch danke ich Inga Konlechner und Robert Kofler für ihre Unterstützung.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Forschungsfragen	5
1.2	Lösungsidee	6
1.3	Beiträge	7
1.4	Einordnung und Abgrenzung	9
1.4.1	Einordnung in die Artefaktororientierung	9
1.4.2	Einordnung in den Softwareentwicklungsprozess	9
1.4.3	Abgrenzung	11
1.5	Erkenntnisprozess	11
1.6	Forschungsmethode und Aufbau der Arbeit	12
2	Grundlagen und verwandte Arbeiten	15
2.1	Domänenwissen	16
2.1.1	Wissen	16
2.1.2	Domäne	25
2.1.3	Domänenwissen	27
2.1.4	Arten von Wissen in der Softwareentwicklung	29
2.2	Modellbildung	30
2.2.1	Modell	31
2.2.2	Modellierung	31
2.2.3	Modell-Semiotik	33
2.2.4	Gemeinsame Modellierung	34
2.2.5	Die Rolle von Modellen in der Softwareentwicklung	36
2.3	Ontologien	38
2.3.1	Definition	39
2.3.2	Gründe für die Verwendung von Ontologien	40
2.3.3	Arten von Ontologien	41
2.3.4	Ontologisches Übereinkommen	44
2.3.5	Entwicklungsprozess	44
2.3.6	Entwicklung	45
2.3.7	Entwicklungs-Kriterien	47
2.3.8	Evaluierung	48
2.4	Verwandte Arbeiten	49
2.4.1	Analysemuster	50
2.4.2	Objektorientierte Analyse	52
2.4.3	Geschäftsprozessanalyse	54
2.4.4	Unternehmensabbildung	57
2.4.5	Zusammenfassung, Stand der Wissenschaft	59

2.5	Zusammenfassung	60
3	Analyse der Überbrückung der Kommunikationslücke in der Praxis	61
3.1	Kontext	61
3.2	Forschungsziel	62
3.3	Aufbau der Studie	62
3.4	Ergebnisse	64
3.4.1	Erfassungsprozess bzw. Modellbildung (A)	65
3.4.2	Stakeholder (B)	69
3.4.3	Management von Artefakten (C)	70
3.4.4	Wiederverwendung (D)	71
3.5	Analyse und Interpretation	72
3.5.1	Forschungsfrage 1: Vorgehensweise in der Praxis	72
3.5.2	Forschungsfrage 3: Auftretende Probleme	73
3.6	Validität der Studie	76
4	Analyse der in der Praxis in Artefakten erhobenen Inhalte	81
4.1	Kontext	81
4.2	Forschungsziel	82
4.3	Forschungsfragen und Hypothesen	82
4.4	Aufbau der Studie	83
4.5	Ergebnisse	84
4.6	Analyse und Interpretation	87
4.7	Validität der Studie	90
5	Interpretation und Ergebnisse, Ableitung einer Lösungsidee	93
5.1	Darstellung der analysierten Probleme	94
5.2	Problem 1: Inhaltliche Missverständnisse	94
5.2.1	Entstehung bzw. Ursache des Problems	95
5.2.2	Beispiel	96
5.2.3	Auswirkungen	97
5.2.4	Mögliche Lösung des Problems	97
5.2.5	Kriterium	99
5.3	Problem 2: Untersch. Abstraktionsebenen von Artefakten	99
5.3.1	Entstehung bzw. Ursache des Problems	99
5.3.2	Beispiel	99
5.3.3	Auswirkungen	100
5.3.4	Mögliche Lösung des Problems	101
5.3.5	Kriterium	103
5.4	Problem 3: Keine Konsistenzsicherung von Artefakten	103
5.4.1	Entstehung bzw. Ursache des Problems	103
5.4.2	Auswirkungen	103
5.4.3	Mögliche Lösung des Problems	104
5.4.4	Kriterium	105
5.5	Problem 4: Unterschiedliche Modellbildung	105
5.5.1	Entstehung bzw. Ursache des Problems	105
5.5.2	Beispiel	105
5.5.3	Auswirkungen	105
5.5.4	Mögliche Lösung des Problems	106

5.5.5	Kriterium	107
5.6	Problem 5: Unausgeglichene Repr. von Wissen	108
5.6.1	Entstehung bzw. Ursache des Problems	108
5.6.2	Beispiel	108
5.6.3	Auswirkungen	108
5.6.4	Mögliche Lösung des Problems	109
5.6.5	Kriterium	110
5.7	Problem 6: Keine Verknüpfung zu Artefakten	111
5.7.1	Entstehung bzw. Ursache des Problems	111
5.7.2	Beispiel	111
5.7.3	Auswirkungen	111
5.7.4	Mögliche Lösung des Problems	112
5.7.5	Kriterium	114
5.8	Problem 7: Kein stakeholderweites Verständnis	114
5.8.1	Entstehung bzw. Ursache des Problems	114
5.8.2	Auswirkungen	115
5.8.3	Mögliche Lösung des Problems	115
5.8.4	Kriterium	116
5.9	Zusammenfassung	116
6	Ontologische Basis	117
6.1	Beispiel	118
6.2	Erfasstes Wissen	120
6.2.1	Konzepte der Problemdomäne	121
6.2.2	Beziehungen zwischen Konzepten der Problemdomäne	123
6.2.3	Allgemeine Konzepte	125
6.2.4	Beziehungen zwischen den allgemeinen Konzepten und den Konzepten der Problem- und Lösungsdomäne	126
6.2.5	Konzepte der Lösungsdomäne	126
6.2.6	Beziehungen zwischen Konzepten der Lösungsdomäne	126
6.2.7	Beziehungen zwischen Konzepten der Problem- und Lösungs- domäne	127
6.3	Formalisierung	128
6.3.1	Problemdomäne	130
6.3.2	Lösungsdomäne	132
6.3.3	Akteure	134
6.3.4	Qualität und Bedingungen	135
6.4	Qualitätseigenschaften eines Basismodells	136
6.4.1	Abgestimmt	136
6.4.2	Angemessen	136
6.4.3	Eindeutig	137
6.4.4	Korrekt	137
6.4.5	Konsistent	137
6.4.6	Modifizierbar	138
6.4.7	Verständlich	138
6.4.8	Vollständig	139
6.5	Zusammenfassung	139
7	Konstruktionsprozess und Verknüpfung zu Artefakten	141

7.1	Einordnung des Modellierungsansatzes	142
7.1.1	Einordnung in den Softwareentwicklungsprozess	142
7.1.2	Einordnung in die Artefaktorientierung	143
7.2	Rollen	145
7.2.1	Modellierer	146
7.2.2	Softwareentwickler bzw. Anforderungsingenieur	146
7.2.3	Domänenexperten	147
7.2.4	Zusammenstellung der Rollen	147
7.3	Konstruktionsprozess	148
7.3.1	Erhebung durch Kompetenzfragen	152
7.3.2	Modellierung: Erweiterung bzw. Anpassung	158
7.3.3	Überprüfung der Konsistenz und Qualität	158
7.3.4	Fertigstellung des Basismodells	158
7.4	Verknüpfung zu anderen Artefakten: Integration	158
7.4.1	Verknüpfung allgemein	159
7.4.2	Verknüpfung zu Anwendungsfällen	160
7.4.3	Verknüpfung zu Sequenzdiagrammen	163
7.4.4	Verknüpfung zu Datenmodellen	166
7.4.5	Verknüpfung zu Komponentendiagrammen	167
7.4.6	Verknüpfung zu Organisationsdiagrammen	169
7.4.7	Verknüpfung zu Geschäftsprozessdiagrammen	171
7.4.8	Verknüpfung zu allgemeinen SOA-Ansätzen	173
7.5	Zusammenfassung	174
8	Evaluation	175
8.1	Industrie- und Praktikumsfallstudien	176
8.1.1	Kontext	176
8.1.2	Forschungsziel	178
8.1.3	Forschungsfragen	178
8.1.4	Studien-Aufbau und Erhebung	180
8.1.5	Ergebnisse Industriestudie	183
8.1.6	Ergebnisse Praktikumsstudie	187
8.1.7	Analyse	189
8.1.8	Validität der Fallstudie	196
8.2	Bewertung des Modells unabhängig von einem Softwareprojekt	198
8.2.1	Kontext	198
8.2.2	Forschungsziel	198
8.2.3	Aufbau und Erhebung	199
8.2.4	Ergebnisse	200
8.2.5	Analyse	202
8.2.6	Validität der Fallstudie	203
8.3	Laborexperiment	203
8.3.1	Kontext	204
8.3.2	Forschungsziel	204
8.3.3	Forschungsfragen	204
8.3.4	Aufbau des Experiments und Erhebung	205
8.3.5	Ergebnisse	208
8.3.6	Analyse	212

8.3.7	Validität des Experiments	213
8.4	Diskussion der Analyse-Ergebnisse	214
8.5	Zusammenfassung	220
9	Zusammenfassung, Ausblick und persönliches Resümee	221
9.1	Ausblick	221
9.1.1	Ontologische Basis und Vorgehen für andere Domänen	221
9.1.2	Anforderungsspezifikation	222
9.1.3	Unterstützung bei der Auswahl von Entitäten für die Verknüpfungen	222
9.1.4	Werkzeugunterstützung	222
9.2	Persönliches Resümee	223
	Literaturverzeichnis	225

Einleitung

Softwaresysteme, eingesetzt im Bankenwesen, Rechnungswesen, Personalverwaltung, usw. werden traditionell in einem aus mehreren Phasen bestehendem Softwareentwicklungsprozess erstellt. Die erste Phase, das Requirements Engineering, ist hinreichend anerkannt die problematischste Phase [LJ00]. Präzise und vollständige Anforderungen sind schwer zu erheben und zusätzlich sind Fehler, die in dieser Phase gemacht werden, in späteren Phasen des Softwareentwicklungsprozesses sehr teuer zu korrigieren.

Im Requirements Engineering sollen die Anforderungen an ein Softwaresystem so spezifiziert werden, dass sie so präzise, vollständig und konsistent wie möglich sind. Anforderungen basieren u.a. auf dem Wissen, das von den Gruppen der Nutzer, Entscheider und Fachexperten seitens des Auftraggebers durch Ermittlungstechniken erhoben wurde.

Die Schnittstelle zwischen Auftraggeber und Auftragnehmer nimmt der Anforderungsingenieur wahr, er versucht das eruierte Wissen in verschiedenen Formen z. B. Text, Tabellen, Grafiken, usw. so präzise und verständlich wie möglich zu erfassen.

Der Anforderungsingenieur ist für gewöhnlich kein Domänenexperte und kann fehlendes Domänenwissen nicht durch die eigene Expertise ersetzen. Daraus resultiert eine Vielzahl von Problemen, u.a. kommt es in der Praxis häufig vor, dass implizites und für Domänenexperten selbstverständliches Wissen nicht explizit erfasst wird und erfasstes Wissen selbst für Domänenexperten der gleichen Domäne eine abweichende Bedeutung haben kann.

Die Gruppe der Entwickler verwendet Anforderungen, um den Entwurf und anschließend die Implementierung durchzuführen. Bereits in der Entwurfs-Phase müssen die vorher ungenau und unpräzise erhobenen Anforderungen von den Entwicklern zum Teil unvollständig interpretiert werden und spätestens in der Implementierungsphase so interpretiert werden, dass sie in einer formalen Sprache abbildbar sind. Die Gruppe der Entwickler trifft daher in diesen Phasen Entscheidungen auf der Basis von meist unzureichendem Wissen über die Domäne des Softwaresystems. In der Software-Entwicklung versucht man diesem Problem mit einer Vielzahl von Ansätzen zu begegnen, u.a. mit frühen Prototypen. Fehlendes Wissen wird dadurch eher identifiziert. Für die Entwicklung von Prototypen werden aber bereits Ressourcen benötigt.

Ausgangssituation. Abbildung 1.1 zeigt die Ausgangssituation. Mehrere Anforderungsingenieure erfassen in frühen Phasen des Requirements Engineerings unterschiedliche Artefakte. Diese werden im Rahmen des Requirements Engineerings konsolidiert in die Anforderungsspezifikation bzw. in das Pflichtenheft integriert und validiert. Die Validierung der Anforderungsspezifikation ist nicht mehr Bestandteil dieser Arbeit und wurde daher bewusst in der Ausgangssituation nicht berücksichtigt. Wir betrachten den Ausschnitt der initialen Erhebung von Problem- und Lösungsdomäne in einem Softwareentwicklungsprozess.

Definition Artefakt:

Unter Artefakt verstehen wir ein Arbeitsergebnis eines Vorganges im Rahmen des Softwareentwicklungsprojektes. Ein Artefakt hat Eigenschaften (Struktur, Verhalten, usw.) und kann mit Hilfe von standardisierten (semi-)formalen Modellierungskonzepten beschrieben sein [MFPKB10]. Wir meinen daher als Artefakt jede Form von expliziertem Wissen (z. B. in Form von Modellen, Dokumenten, handschriftliche Skizzen, usw.) als Ergebnis eines Erfassungsvorganges. Der Erfassungsvorgang kann z. B. aufgrund eines Erhebungsvorganges angestoßen werden.

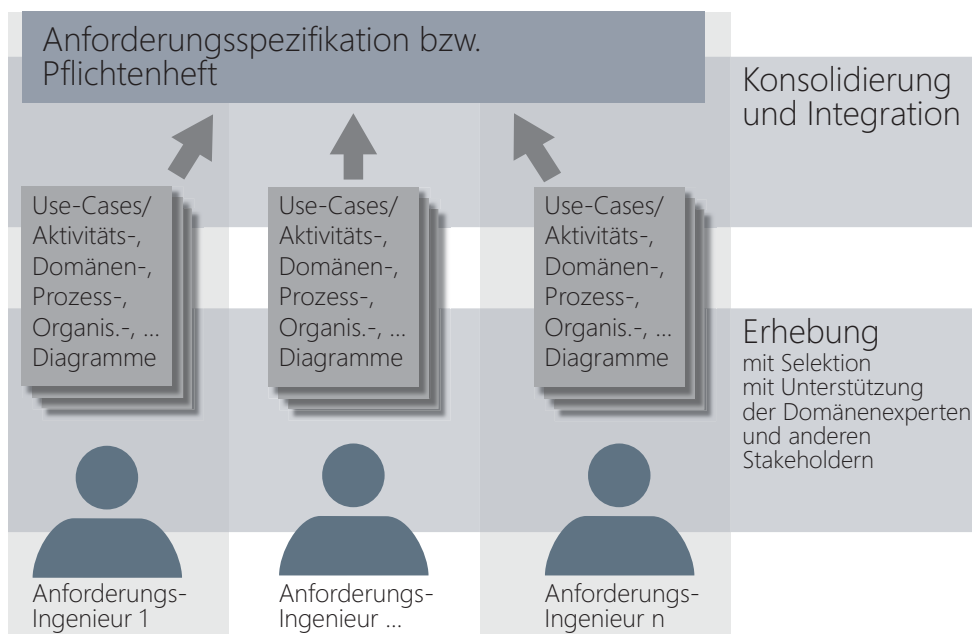


Abbildung 1.1: Ausgangssituation in frühen Phasen des Requirements Engineerings

Gerade die im Rahmen der Erhebung erfassten Artefakte können häufig nicht zueinander in Beziehung gesetzt werden. Bei der Integration der Artefakte in die Anforderungsspezifikation bzw. dem Pflichtenheft müssen Interpretationslücken geschlossen werden bzw. inhaltliche Widersprüche identifiziert und behoben werden. Dies ist bereits bei kleineren Projekten ein Problem, da es zumeist keine Vorgaben gibt, z. B. in welcher Abstraktionsebene Artefakte erfasst werden. Beispielsweise wird ein Manager, wenn dieser über einen konkreten Geschäftsprozess spricht, in einer höheren Abstraktionsebene darüber sprechen, als ein Prozessverantwortlicher. Wenn diese unterschiedlichen Darstellungen des Geschäftsprozesses zusammengefügt werden, müssen eventuell entstandene Interpretationslücken geschlossen werden.

Im Rahmen dieser Arbeit haben wir Probleme identifiziert die in Artefakten, erstellt in frühen Phasen des Requirements Engineerings, auftreten. Diese werden in Kapitel 5 im Detail behandelt.

Inhaltliche Missverständnisse. In Artefakten, welche in frühen Phasen des Requirements Engineerings erstellt wurden, wird unterschiedliches Wissen erfasst. Im Gegensatz zu beispielsweise im Rahmen vom Feindesign erstellten Klassendiagrammen, wird nicht nur lösungsdomänenspezifisches Wissen erfasst, sondern vor allem (zum Fördern des Problemverständnisses) problemdomänenspezifisches Wissen. Problemdomänenspezifisches Wissen weicht von seiner Abbildung in der Lösungsdomäne häufig ab, z. B. werden bestimmte Eigenschaften nicht in der Software abgebildet, oder, es werden Eigenschaften in der Software abgebildet, die nicht Teil der Problemdomäne sind. Dies führt zu Missverständnissen beim Interpretieren der Modelle durch Entwickler in späteren Phasen und zu Missverständnissen bei der Erhebung der Problemdomäne im Requirements Engineering, da die unterschiedliche Repräsentation eines in der Problemdomäne vorhandenen abgebildeten Originals von diesem abweicht.

Definition Problemdomäne:

Unter Problemdomäne verstehen wir ein abgrenzbares Problemfeld oder den Anwendungsbereich für ein Softwaresystem.

Definition Lösungsdomäne:

Unter Lösungsdomäne verstehen wir einen abgrenzbaren Bereich, der sich nicht mit der Problemdomäne überschneidet und all das umfasst, was direkt zur Lösung eines Problems in der Problemdomäne beiträgt.

Artefakte auf unterschiedlichen Abstraktionsebenen. Abhängig vom Wissensträger werden Problemdomänen in unterschiedlichen Abstraktionsebenen erfasst. Z. B. wird ein für mehrere Arbeitsbereiche verantwortlicher Manager einen bestimmten Arbeitsbereich auf einer anderen Abstraktionsebene erklären, als es eine für diesen konkreten Arbeitsbereich verantwortliche Person tun würde. Dies liegt u.a. daran, dass der Manager eine andere Sichtweise auf diesen Arbeitsbereich hat (z. B. eine integrative Sicht über mehrere Arbeitsbereiche hinweg).

Typischerweise sind beide Sichtweisen für ein zu entwickelndes Softwaresystem von Bedeutung. Selbige müssen wieder integriert werden, damit vorhandene Interpretationslücken mittels der Integration geschlossen und Widersprüche erkannt werden können. Dies ist schon deswegen häufig schwierig, da zwischen den geschilderten Abstraktionsebenen eine Lücke klafft, die nicht einfach durch zusätzliche Beziehungen (z. B. Detaillierung) ausgedrückt werden kann.

Keine Konsistenzsicherung von Artefakten. Artefakte werden in frühen Phasen des Requirements Engineerings ohne Vorgaben erstellt. Die erstellten Artefakte sind typischerweise semi-formale Modelle (z. B. auf Basis von UML-Klassendiagrammen) oder auf Basis von Textschablonen (z. B. Use-Cases/Szenarien). Dies liegt daran, dass in diesen initialen Artefakten noch keine Formalismen eingesetzt werden, da das Er-

arbeiten einer gemeinsamen Vorstellung von Problem und Lösung im Mittelpunkt steht. Die in dieser Phase entstandenen Artefakte sind häufig untereinander inkonsistent und können nur sehr aufwändig (manuell) auf ihre Konsistenz überprüft werden.

Unterschiedliche Modellbildung. Artefakte werden von unterschiedlichen Personen unterschiedlich erstellt. Die Modellbildung ist von subjektiven Einflüssen geprägt. U.a. aufgrund unterschiedlicher Sichten auf eine Problemdomäne, bzw. unterschiedlicher Prioritäten bei der Modellbildung. Z. B. wird ein Geschäftsprozess in einer Modellierungstechnik von einem Modellierer als eigene Entität dargestellt und von einem anderen als benannte Eigenschaft zwischen zwei Entitäten.

Dies führt (a) zu Verständnisschwierigkeiten beim Lesen bzw. Interpretieren der Modelle, da scheinbar gleiche Sachverhalte unterschiedlich dargestellt werden und (b) wird so die Integration der Modelle in der Anforderungsspezifikation schwieriger.

Unterschiedliche Prioritäten bei der Modellbildung. Ein Grund für die unterschiedliche Modellbildung, wie im vorherigem Absatz erwähnt, sind die unterschiedlichen Prioritäten der Modellierer. Dies führt dazu, dass manche Sachverhalte in Artefakten mit einer höheren Priorität, und dadurch auch detaillierter, erfasst werden, als andere Sachverhalte. Die Prioritäten werden dabei typischerweise nach subjektiven Gesichtspunkten gesetzt. Z. B. wird eine arbeitsbereichverantwortliche Person seinen eigenen Arbeitsbereich mit einer subjektiv höheren Wichtigkeit einstufen als andere Arbeitsbereiche. Dies führt u.a. dazu, dass viele Details zu einem subjektiv wichtigem Arbeitsbereich erfasst wurden, aber ein womöglich ebenso wichtiger Arbeitsbereich nur rudimentär erfasst wurde.

Keine Verknüpfung zwischen den Artefakten. In frühen Phasen des Requirements Engineerings werden unterschiedliche Artefakte erstellt. Es gibt keine expliziten Vorgaben zur Integration der Artefakte.

Artefakte werden in der Anforderungsspezifikation bzw. im Lastenheft integriert. Diese Integration wird häufig nur textuell in den Artefakten beschrieben, in denen die Artefakte integriert wurden. Einen Zusammenhang zwischen den erstellten Artefakten zu identifizieren und zu beschreiben, ist nur mit sehr viel Aufwand übergreifend möglich. Diese fehlende Integration führt u.a. zu inkonsistenten Artefakten.

Kein stakeholderweites Verständnis. Einer der Ursachen der angeführten Probleme und Sachverhalte ist das fehlende einheitliche Verständnis über die Problem- domäne seitens den Anforderungsingenieuren, sowie dem fehlendem Verständnis der Lösungsdomäne, seitens der Stakeholder der Verwender bzw. Auftraggeber.

Definition Gesamtverständnis:

In Kontext dieser Arbeit ist das Gesamtverständnis das gemeinsame Verständnis aller relevanten Stakeholder über die Problem- und Lösungsdomäne auf einer gemeinsam festgelegten Abstraktionsebene.

1.1 Forschungsfragen

Im Rahmen dieser Arbeit stellen wir vier Forschungsfragen. Die Motivation zur Forschung in diese Richtung kommt aus der Praxis.

In der Literatur wird häufig die Kommunikationslücke aufgrund fehlendem Domänenverständnisses seitens der Softwareentwickler erwähnt [CKI88, GK95, LJ00, Off02, EW05]. Sie ist für viele Probleme in späteren Phasen mitverantwortlich, z. B. weil bei nicht spezifizierten Sachverhalten Softwareentwickler Löcher in der Spezifikation durch Interpretation schließen.

In der Praxis wird Software im Auftrag entwickelt. Die Kommunikationslücke zwischen Auftraggeber und Auftragnehmer muss dabei überbrückt werden. Im Rahmen dieser Arbeit wollten wir erheben, wie die Kommunikationslücke¹ überbrückt wird, welche Inhalte erhoben werden und welche Probleme bezüglich der Überbrückung auftreten können.

Daraus haben wir vier Forschungsfragen formuliert, die wir im Folgendem erläutern:

RQ1: Wie wird die Kommunikationslücke in Softwareprojekten in frühen Phasen des Requirements Engineerings in der Praxis überbrückt?

Artefakte werden in frühen Phasen des Requirements Engineerings ohne Vorgaben erstellt. Sie dienen dem Erarbeiten eines gemeinsamen Verständnisses. Diese Artefakte sind typischerweise Grundlage für detailliertere u.a. Modelle (z. B. Aktivitäts- Interaktionsdiagramme, E/R-Modelle, ...) und sie dienen der initialen Überbrückung der Kommunikationslücke.

RQ2: Welche Inhalte werden in der Praxis in Artefakten erfasst, die zur Überbrückung der Kommunikationslücke in frühen Phasen des Requirements Engineerings erstellt wurden?

Gerade in frühen Phasen werden typischerweise problemdomänenspezifische Inhalte erfasst. Typische Modellierungstechniken sind E/R-Diagramme, Prozesslandschaften, Abläufe, usw.

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

Die in RQ2 erhobenen Inhalte führen zu Problemen, die wir in Abschnitt 5 analysiert werden. Viele der Probleme lassen sich darauf zurückführen, dass es keine Möglichkeit gibt, diese verschiedenen Sichten *einfach* zu integrieren, weil sie z. B. auf verschiedenen Abstraktionsebenen erfasst wurden oder durch die verschiedenen Modellierer auch verschiedene subjektiv wichtige Sachverhalte enthalten.

Basierend auf die durch die Beantwortung von RQ3 gefundenen Probleme haben wir eine Modellierungstechnik erarbeitet, die diese Probleme adressiert und so Forschungsfrage RQ4 formuliert.

¹ Unter Kommunikationslücke meinen wir auftretende Probleme bei der Kommunikation in Softwareprojekten zwischen Auftraggebern und Auftragnehmern.

RQ4: Inwiefern hilft der erarbeitete Ansatz bei der Lösung der Probleme bezüglich RQ3 bei der Erhebung von Problem- und Lösungsdomäne in frühen Phasen des Requirements Engineerings?

Im Rahmen der Beantwortung der Forschungsfrage RQ4 in der Evaluation, betrachten wir, inwiefern die erarbeitete Modellierungstechnik dabei hilft, die durch die Beantwortung von RQ3 gefundene Probleme zu adressieren.

1.2 Lösungsidee

Wir schlagen im Rahmen der in der Problemanalyse (siehe Kapitel 5) erhobenen Probleme einen Ansatz vor, der beim Erheben der Problem- und Lösungsdomäne in frühen Phasen des Requirements Engineerings als Unterstützung dient.

Dazu wird in einer kollaborativen Tätigkeiten, in die verschiedene Stakeholder involviert sind, ein Modell erarbeitet. Dieses Modell expliziert Wissen über die Problem- und Lösungsdomäne und verknüpft dieses untereinander.

Das erarbeitete Modell dient als Basis für weitere Artefakte. Dieses Modell nennen wir in dieser Arbeit Basismodell.

Definition ontologische Basis:

Eine ontologische Basis enthält das terminologische Wissen einer bestimmten Domäne. Terminologisches Wissen besteht aus Konzepten und deren möglichen Beziehungen untereinander. Zusätzlich enthält die ontologische Basis Axiome.

Definition Basismodell:

Unter Basismodell verstehen wir eine Ontologie, die auf der Terminologie der ontologischen Basis basiert. Daher sind alle Entitäten im Basismodell durch die Zuweisung eines Konzeptes und alle Beziehungen durch die Zuweisung einer der möglichen Beziehung zwischen den Konzepten stereotypisiert.

Abbildung 1.2 zeigt die konzeptuelle Lösungsidee der in der Einleitung angesprochenen Probleme. Anforderungsingenieure erfassen mit Hilfe der Domänenexperten das Basismodell, welches das Gesamtverständnis auf einer bestimmten Abstraktionsebene enthält. Das Basismodell dient als Ausgangspunkt (Basis) für die Detaillierung in anderen Modellierungstechniken.

Im Unterschied zur Ausgangssituation in Abbildung 3.1 wird vor dem Erstellen der Artefakte, die typischerweise in frühen Phasen des Requirements Engineerings erstellt werden, das Basismodell erstellt. Das Basismodell enthält auf einer hohen Abstraktionsebene Entitäten, die das Gesamtverständnis repräsentieren.

Auf Basis des Basismodells werden alle weiteren Artefakte erstellt. Diese Vorgehensweise hat positive Auswirkungen auf die Qualität und Konsistenz der Artefakte, welche auf dem Basismodell basieren. In weiterer Folge ist es einfacher die erstellten Artefakte in die Anforderungsspezifikation bzw. in das Lastenheft zu integrieren.

Der Zweck des Basismodell ist es, (a) inhaltliche Missverständnisse zu reduzieren, (b) als Basis für andere Artefakte dafür zu sorgen, dass sich diese Artefakte auf ei-

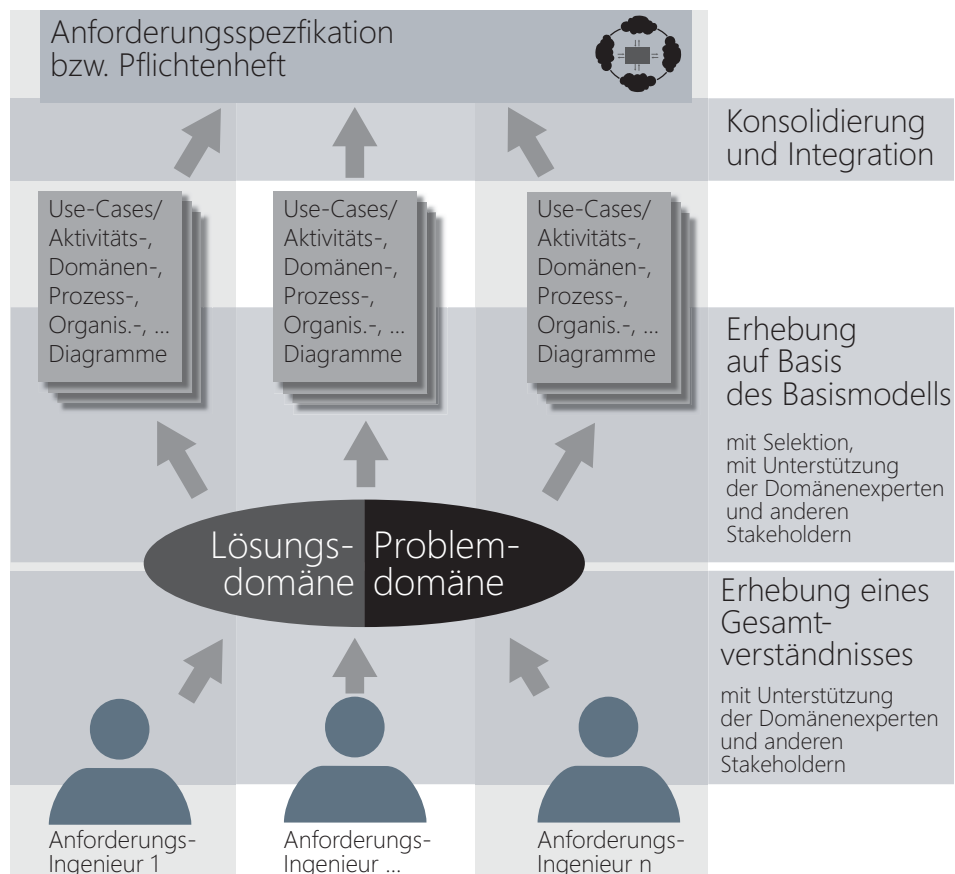


Abbildung 1.2: Situation in frühen Phasen des Requirements Engineerings nach Anwendung der Lösung

ner ähnlichen Abstraktionsebene befinden, (c) die Konsistenz der darauf basierenden Artefakte untereinander zu verbessern, (d) die Modellbildung so zu lenken, dass ähnliche Sachverhalte auch ähnlich repräsentiert werden, (e) eine ausgeglichene Repräsentation von Wissen in allen auf dem Basismodell basierenden Artefakten zu erreichen, (f) die Möglichkeit der Verknüpfung zu anderen Artefakt-Typen zu bieten und (g) ein stakeholderweites Verständnis zu repräsentieren.

1.3 Beiträge

Diese Arbeit liefert folgende Beiträge zum Stand der Technik und Praxis:

- **Problemanalyse:** Die Arbeit leistet einen Beitrag in der Explizierung von den in diesem Kapitel anfangs erwähnten und in Kapitel 5 diskutierten Problemen, die in frühen Phasen des Requirements Engineerings auftreten und bisher nur wenig bis keine Beachtung in der Literatur fanden.

Hierzu wurden einerseits Tiefeninterviews durchgeführt, um herauszufinden, wie die Erhebung von Anforderungen in der Praxis vonstatten geht und wie die Kommunikationslücke überbrückt wird. Daraus extrahieren wir eine Reihe von Problemen, die wir mit der Analyse von Artefaktinhalten (von Artefak-

ten, die in frühen Phasen des Requirements Engineerings erstellt wurden) noch weiter untermauern.

- ▶ **Modellierungssprache:** Basierend auf den Inhalte von Artefakten, die in frühen Phasen des Requirements Engineerings erstellt wurden, schlagen wir eine minimalistische Modellierungssprache vor. Mit Hilfe dieser Sprache wird in frühen Phasen des Requirements Engineerings ein Modell erstellt – das Basismodell, auf dem andere Artefakte basieren. Alle darauf aufbauenden Artefakte müssen zu den Inhalten, die im Basismodell erfasst wurden, konform sein. Dies führt zu besser strukturierten, konsistenten und besser verständlichen Artefakten.

Das Basismodell repräsentiert ein Gesamtverständnis aus Problem- und Lösungsdomäne. Weiterer Teile der Modellierungssprache sind:

- **Ableitungs- und Konsistenzregeln:** Zusätzlich zur Modellierungssprache stellen wir Konsistenz- und Ableitungsregeln vor, die wir axiomatisch spezifiziert haben. Diese beruhen auf Rückmeldungen von Anwendern sowie auf der Analyse von Basismodellen und Anmerkungen von Kollegen.

Zweck der Ableitungs- und Konsistenzregeln ist das Sicherstellen der Konsistenz des Basismodells und folglich auch der darauf basierenden Artefakte.

- **Qualitätseigenschaften:** Basierend auf den Qualitätseigenschaften von Anforderungen und Ontologien haben wir die Qualitätseigenschaften des Basismodells expliziert.
- ▶ **Methodik:** Die Methodik basiert auf Ansätzen, welche in der Domänenanalyse und Ontologieentwicklung erarbeitet wurden. Darauf aufbauend haben wir einen Prozess definiert, der den Konstruktionsprozess darstellt. Dieser besteht aus einer Menge von Aktivitäten. Innerhalb von Aktivitäten zur Erhebung des initialen Gesamtverständnisses, wird mit Kompetenzfragen² gearbeitet. Diese Fragen helfen beim Identifizieren von relevantem Wissen.

Diese Vorgehensweise erlaubte es uns einen Erhebungsdiallog zwischen den verschiedenen Rollen, die in der Erstellung eines Basismodells beteiligt sind, zu formulieren, der eine strukturierte Erstellung des Basismodells ermöglicht.

Beispielhaft wird im Rahmen der Methodik gezeigt, wie eine Verknüpfung des Basismodells zu den Inhalten anderer Artefakt-Typen durchgeführt werden kann.

- ▶ **Evaluation:** Zur Evaluierung des Modellierungsansatzes wurden einige Fallstudien und ein Laborexperiment durchgeführt. Der Modellierungsansatz wurde in einer Fallstudie in der Industrie erprobt. Aufbauend auf die dadurch gewonnen Erkenntnisse wurde der Modellierungsansatz angepasst und in verschiedenen App-Entwicklungsprojekten eingesetzt. In beiden Fallstudien fand eine Bewertung durch die Ersteller des Basismodells, sowie jenen Teams statt,

² Eine Kompetenzfrage ist im Rahmen der Ontologieentwicklung eine Frage in natürlicher Sprache, die dabei hilft (a) das Scoping einer Ontologie festzulegen und (b) die Ontologie zu evaluieren [AGP04]. Eine Kompetenzfrage muss durch die entwickelte Ontologie beantwortbar sein. Die Definition befindet sich auf Seite 148.

die das Basismodell als Basis für weitere Artefakte verwendeten. Zur Feststellung von Eigenschaften des Basismodells, die nicht im Rahmen der Fallstudien untersucht werden konnten, wurde ein Laborexperiment durchgeführt.

1.4 Einordnung und Abgrenzung

In diesem Abschnitt nehmen wir die Einordnung des Ansatzes in die Artefaktorientierung (Abschnitt 1.4.1) und in den Softwareentwicklungsprozess (Abschnitt 1.4.2) vor. Abschließend folgt eine Abgrenzung (Abschnitt 1.4.3).

1.4.1 Einordnung in die Artefaktorientierung

Die Idee hinter der Artefaktorientierung ist es, sich bei der Entwicklung eines Softwaresystems auf die zu erstellenden Artefakte zu konzentrieren und nicht auf die anzuwendenden Methoden oder Prozessschritte [FLPW11].

Im Rahmen einer artefaktbasierten Vorgehensweise wird ein Referenzmodell erstellt, welches alle Artefakte enthält, die im Laufe des Entwicklungsprojektes zu erstellen sind. Hierzu werden die Artefakte, die Beschreibungsmittel und die Struktur bestimmt. Im Projekt wird ein Prozess definiert, indem man sich einigt, welche Rollen, welche Artefakte erstellen und festlegt, bis zu welchen Meilenstein dies geschehen soll.

Méndez Fernández et al. [FLPW11] haben im Rahmen eines Industrieprojektes ein Referenzmodell erstellt, das sie BISA (*business information systems analysis*) nennen.

Das im Referenzmodell vorkommende Artefakt *System Vision* dient u.a. dem Scoping³. Nur ein Teil des in Artefakten explizierten Wissens wird in den nächsten Abstraktionsebenen benötigt [Fer11]. Bevor detailliert Anforderungen erhoben werden, wird der Anwendungsbereich des Systems mit dem Artefakt *System Vision* definiert. Die *System Vision* muss mit allen relevanten Stakeholdern abgestimmt sein [Fer11].

Diese Arbeit schlägt eine Ausprägung des Artefaktes *System Vision* vor. Wir geben vor, (a) wie die Struktur dieses Artefaktes aufgebaut ist, (b) welche Inhalte erfasst werden, (c) wie die Erhebung des Artefaktes vorgenommen werden soll, (d) wie die Konsistenz des Artefaktes sichergestellt werden kann und (e) wie eine mögliche Verknüpfung zu anderen Artefakt-Typen (z. B. Anwendungsfällen) konkret vorgenommen werden kann.

1.4.2 Einordnung in den Softwareentwicklungsprozess

Ein typischer Softwareentwicklungsprozess besteht zumindest aus den Phasen Requirements Engineering, Entwurf, Implementierung und Testen, dies wird auf der linken Seite der Abbildung 1.3 dargestellt. Der hier vorgestellte Ansatz wird im Requirements Engineering angewendet. Requirements Engineering besteht typischer-

³ Scoping leitet sich aus dem Englischen (*scope*) ab und bedeutet soviel wie die Festlegung eines Geltungsbereiches bzw. das Abgrenzen eines Bereiches.

weise aus den Phasen Erhebung, Analyse/Modellierung, Spezifikation und Validierung [AW05], dies wird auf der rechten Seite der Abbildung 1.3 dargestellt.

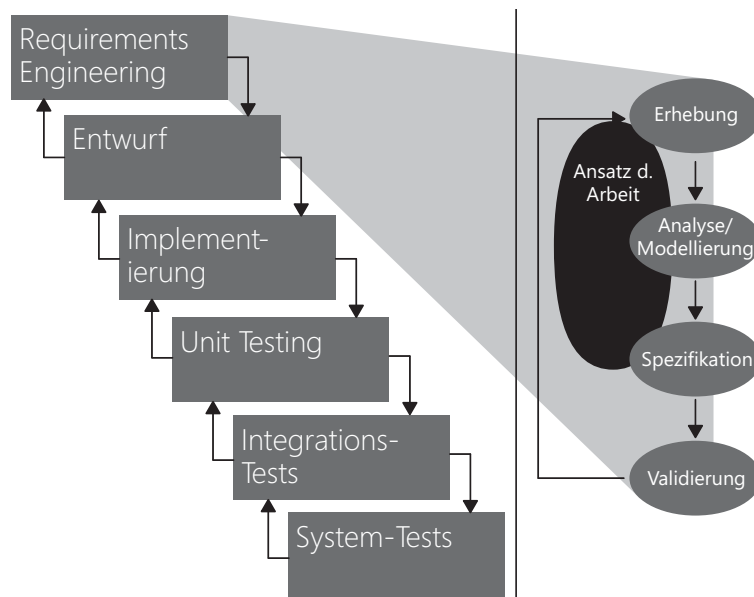


Abbildung 1.3: Einordnung in den Softwareentwicklungsprozess und dem Requirements Engineering

- ▶ **Erhebung:** Erhebung von Anforderungen durch die Kommunikation mit den relevanten Stakeholdern des Auftraggebers (z. B. Benutzer, Domänenexperten).
Beitrag der Arbeit: Die Erhebung einer gemeinsamen Vorstellung des Problems, sowie Anforderungen an eine mögliche Lösung, wird durch einen vorgegeben Erhebungsprozess zwischen den verschiedenen involvierten Stakeholdern gesteuert. Das Produkt dieses Prozesses ist ein Modell (das Basismodell), das diese Vorstellung über Problem und Anforderungen an die Lösung abbildet. Dieses Basismodell wird in den weiteren Phasen des Requirements Engineering an verschiedenen Stellen eingesetzt.
- ▶ **Analyse/Modellierung:** Ausarbeitung von Modellen, die unterschiedliche Sichtweisen auf die Problemdomäne repräsentieren. In der Analyse wird festgelegt, welche Ausschnitte der modellierten Sichten Ausschnitte der Umgebung des zu entwickelnden Systems sein sollen.
Beitrag der Arbeit: Die Verknüpfung des im Rahmen der Anwendung des Modellierungsansatzes erstellten Basismodells zu anderen Artefakten, wird im Rahmen der Arbeit beispielhaft für wichtige Artefakt-Typen des Requirements Engineerings gezeigt.
- ▶ **Spezifikation:** Einzelne Modelle bzw. Modellelemente werden in diesem Schritt präzise beschrieben.
Beitrag der Arbeit: Auf Basis des Basismodells entstehen durch die Verknüpfung zu anderen Artefakt-Typen besser strukturierte Artefakte. Der Zusammenhang zwischen den Artefakten wird durch das durch den Modellierungsansatz erstellte Basismodell expliziert.
- ▶ **Validierung:** Kontrolle der Korrektheit, Vollständigkeit und Konsistenz. Es wird die Frage beantwortet, ob das *richtige* System gebaut wird?

1.4.3 Abgrenzung

Der in dieser Arbeit vorgeschlagene Modellierungsansatz enthält Konzepte einer Vielzahl von bereits vorhandenen Modellierungstechniken, u.a. Geschäftsprozess-, Organisations-, Komponenten- und Datenmodellierung.

Im Rahmen dieser Arbeit soll keine bereits vorhandene Modellierungstechnik ersetzt werden. Der Autor sieht den Ansatz als Ergänzung zu bereits vorhanden Modellierungstechniken. Der Ansatz zielt darauf ab, ein frühes Gesamtverständnis über die Problem- und Lösungsdomäne zu erarbeiten. Auf Basis dieses Gesamtverständnisses werden dann weitere Artefakte mit den anfänglich erwähnten Modellierungstechniken erstellt, bzw. das Gesamtverständnis beruht zum Teil auf bereits vorhandenen Artefakte. Dieses Gesamtverständnis, in Form eines Modells (dem Basismodell), dient als konsistente Basis, alle darauf aufbauenden Artefakte dürfen den Aussagen, expliziert im Basismodell, nicht widersprechen.

1.5 Erkenntnisprozess

Für die wissenschaftliche Vorgehensweise wurde der Leitfaden in [sWB10] verwendet. Wir haben die Arbeiten an die Phasen der Vorgehensweise ausgerichtet.

In dieser Arbeit verwenden wir den epistemologischen Bezugsrahmen von Becker et al. [BHK03, BNK04]. Wir ordnen uns dem epistemologischen Bezugsrahmen, mit seinen Bestandteilen: Gegenstand der Erkenntnis, Relation Erkenntnis zum Erkenntnisgegenstand, wahre Erkenntnis, Quelle der Erkenntnis und Mittel der Erkenntnis entsprechend ein.

Erkenntnisgegenstand – Ontologischer Aspekt. Becket et al. [BNK04] differenzierten in der Forschung verschiedene Positionierungen: Der Autor dieser Arbeit geht von der Untersuchung der Realwelt aus, die unabhängig vom menschlichen Bewusstsein ist. Daher nehmen wir die Wirklichkeit als ein vom menschlichen Bewusstsein unabhängiges Konstrukt wahr. Wir nehmen daher die Position des ontologischen Realismus ein [BNK04].

Verhältnis von Erkenntnis und Erkenntnisgegenstand. In der Frage, ob sich Erkenntnis und Erkenntnisgegenstand außerhalb des menschlichen Denkens und Sprechens zumindest prinzipiell objektiv erkennen lassen, folgen wir dem *Konstruktivismus*. Die Erkenntnis erfolgt zumindest initial *privat* [BNK04]. Das Verhältnis zwischen der Erkenntnis und dem Erkenntnisgegenstand ist somit entscheidend durch das erkennende Subjekt geprägt [BNK04].

Wahre Erkenntnis. Wahrheit in dieser Arbeit folgt der Definition, dass eine Aussage genau dann wahr für eine Gruppe ist, wenn sie unter idealen und optimalen Bedingungen für die Gruppe akzeptierbar ist [BNK04]. Eine Aussage ist daher dann wahr, wenn sie im Rahmen einer Wahrheitsprüfung für die Gruppe als wahr ersichtlich ist [BNK04].

Quelle der Erkenntnis. Unter Quelle der Erkenntnis verstehen wir die Beantwortung der Frage, woher das Wissen zur Erkenntnis stammt. Als Wissensquellen gelten Erfahrung (das auf Erfahrungen basiertes Wissen, auch aposteriorisches oder empirisches Wissen), der Verstand bzw. Intellekt (durch begriffliche Bemühungen des Subjekts unter Verwendung eines Unterscheidungssystems Gegenstand der Erkenntnis werdend) und sowie aus beiden [BNK04]. In allen weiteren Kapiteln dieser Arbeit nutzen wir beide Erkenntnisquellen.

Mittel der Erkenntnis – methodologischer Aspekt. Der Weg der Herleitung von neuem Wissen bezeichnet den methodologischen Aspekt. Die Erkenntnis kann zum einem *induktiv* (das Schließen von Einzelfällen auf allgemeingültige Sätze – daher die Verallgemeinerung) und zum anderem *deduktiv* (der Ableitung einer Aussage aus anderen Aussagen kraft logischer Schlussregeln – daher das Ableiten des Einzelnen aus dem Allgemeinen) sein [BNK04, RH09]. Diese Arbeit verwendet zur Ableitung der Problemstellung (Problemanalyse) den deduktiven Erkenntnisgewinn. Die Evaluation bzw. Anpassung des Modellierungsansatzes erfolgte unter Anwendung von induktiven Vorgehensweisen.

1.6 Forschungsmethode und Aufbau der Arbeit

Österle et al. [sBF⁺10, sWB10] formulierten Phasen zum iterativen Vorgehen bzgl. des Erkenntnisprozesses, der sich auch in Abbildung 1.4 manifestiert. Im Folgenden wird auf die Schritte in [sBF⁺10] eingegangen und ihre Repräsentation in dieser Arbeit erläutert:

► **Problemanalyse:**

- **Studien:** Der Anstoß eines Forschungsthemas kann aus der Wissenschaft und aus der Praxis kommen [sBF⁺10]. In unserem Fall kommt der Anstoß vor allem aus der Praxis. Siehe dazu Kapitel 3.

In der Wissenschaft finden die erhobenen Probleme derzeit in frühen Phasen des Requirements Engineerings keine direkte Beachtung. Sehr wohl gibt es Ansätze, die angesprochenen Probleme aus anderen Perspektiven betrachten, aber nur Teilprobleme dabei adressieren. Siehe dazu Abschnitt 2.4.

- **Problemidentifikation:** Die Probleme werden in Kapitel 3 auf Basis des Standes der Praxis erhoben und in Kapitel 4 auf Basis der Inhalte in erstellten Artefakte. Zusammengeführt wird dies in Kapitel 5. Dieses Kapitel schlägt auch für die jeweiligen Probleme einen Lösungsansatz vor.
- **Forschungsziele:** Die Forschungsziele ergeben sich aus den beantworteten Forschungsfragen, primär in Kapitel 5. Forschungsziel ist es, einen Beitrag zur Lösung der in Kapitel 5 im Detail beschriebenen Probleme zu leisten.

► **Konstruktive Forschung:**

- **Ontologische Basis:** Auf Basis der Inhalte, erhoben in Artefakten in der

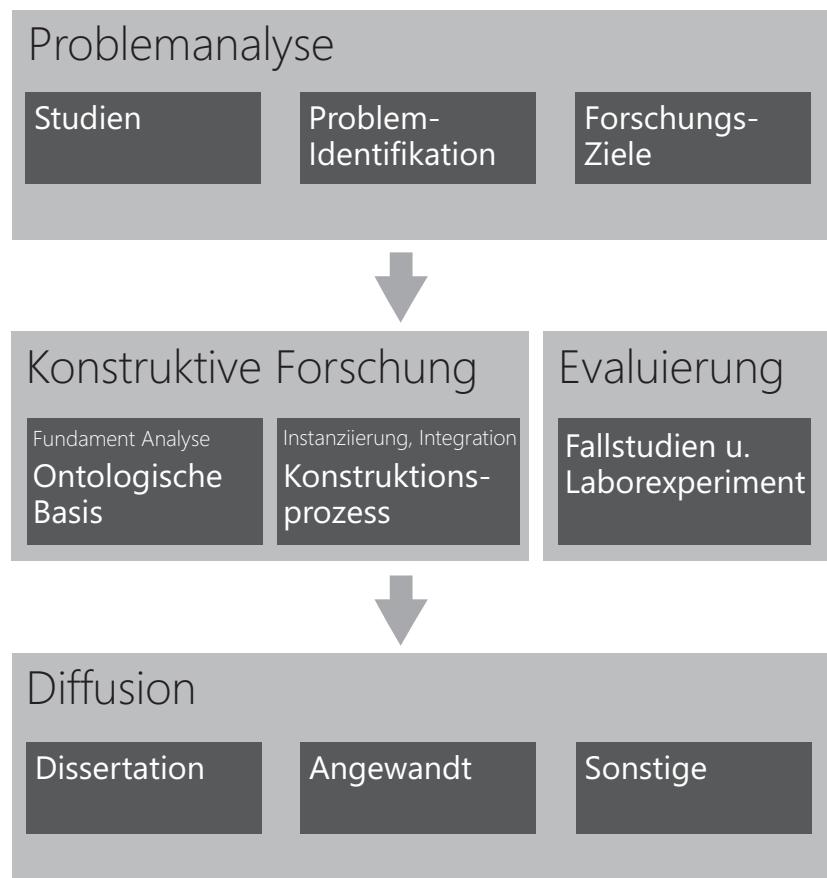


Abbildung 1.4: Angewandte Forschungsmethode

Praxis, die wir der Enterprise Ontologie von Uschold et al. [UKMZ98] zugeordnet haben, schlagen wir eine Ontologie vor, die die in den verschiedenen Artefakten erhobenen Inhalte repräsentieren kann. Auf Basis der Beziehungsarten haben wir die Beziehungen zwischen den Konzepten⁴ der Ontologie abgeleitet und Ableitungs- und Konsistenzregeln aufgestellt. Die ontologische Basis wurde in verschiedenen Projekten angewendet und verbessert bzw. erweitert und/oder auch reduziert. Dies wird in Kapitel 6 erläutert.

- **Konstruktionsprozess:** Basierend auf die getane Forschung im Bereich Entwicklung von Ontologien bzw. der Domänenanalyse, haben wir den Konstruktionsprozess formuliert. Der Konstruktionsprozess wurde in verschiedenen Projekten angewendet und angepasst. Nach der Konstruktion eines Basismodells wird dieses weiter verwendet. Diese Weiterverwendung wird ebenfalls exemplarisch gezeigt. Erläutert wird dies in Kapitel 7.

► **Evaluation:**

⁴ Ein ontologisches Konzept (in dieser Arbeit kurz Konzept) beschreibt eine Menge von Eigenschaften, die von einer oder mehreren Entitäten erfüllt sein können. Ein ontologisches Konzept fasst daher Eigenschaften zusammen und entspricht einer Vorstellung von etwas, bei dem die Eigenschaften bekannt sind. Die Definition befindet sich auf Seite 117.

- **Fallstudie Industrie:** Der Ansatz wurde in einem Softwareentwicklungsprojekt eingesetzt und abschließend vom Team bewertet, dies wird in Abschnitt 8.1 erläutert.
 - **Fallstudie App-Entwicklung:** Im Rahmen der Entwicklung von Apps für Smartphones und Tablets wurde der Ansatz in sechs Entwicklungsprojekten eingesetzt. Diese werden ebenfalls Abschnitt 8.1 erläutert.
 - **Laborexperiment:** Nicht alle Kriterien der in Kapitel 5 im Detail beschriebenen Probleme konnten im Rahmen der Fallstudien evaluiert werden. Ergänzend wurde daher ein Laborexperiment durchgeführt. Dies wird in Abschnitt 8.3 erläutert.
- **Diffusion:**
- **Dissertation:** Diese Arbeit ist primäres Artefakt zur Verbreitung des Wissens, das im Rahmen dieser Arbeit geschaffen wurde, in der wissenschaftlichen Gemeinschaft.
 - **Angewandt:** Die Ideen und Methoden, die in Rahmen dieser Arbeit entwickelt wurden, wurden durch Workshops einerseits in verschiedenen Unternehmen verbreitet und andererseits im Rahmen der Lehrtätigkeiten eingebracht.

Grundlagen und verwandte Arbeiten

Dieses Kapitel führt in die grundlegenden Begriffe und Grundlagen ein, die zum Verstehen des erarbeiteten Modellierungsansatzes notwendig sind.

Schwerpunkt dieses Kapitels ist die Einführung von Begrifflichkeiten, sowie die Darstellung von grundlegenden Einflussfaktoren auf den Modellierungsansatz.

In frühen Phasen des Requirements Engineerings wird typischerweise versucht ein Problemverständnis zu erarbeiten. Hierzu ist es notwendig, das relevante Wissen über eine Domäne zu verstehen und zu explizieren. Hierzu nähern wir uns dem Domänenwissen, indem wir den Wissens-Begriff in Abschnitt 2.1.1 erläutern. Der Begriff Domäne wird vielfach in der Softwareentwicklung verwendet, wir beschäftigen uns mit diesem Begriff in Abschnitt 2.1.2. In Abschnitt 2.1.3 führen wir den Wissens-Begriff mit dem Domänen-Begriff zusammen.

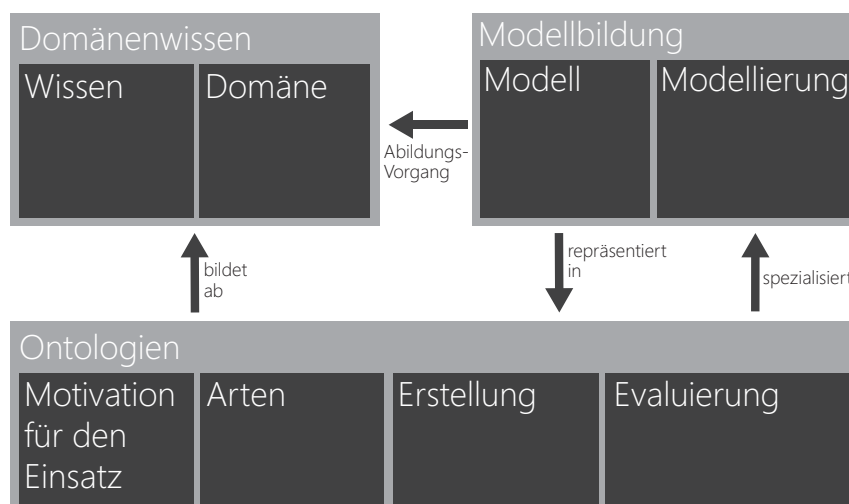


Abbildung 2.1: Aufbau der Abschnitte der Grundlagen

Abbildung 2.1 zeigt den Aufbau dieses Kapitels. Typischerweise werden in frühen Phasen des Requirements Engineerings Modelle erstellt, wir beschäftigen uns in Ab-

schnitt 2.2.1 mit dem Modell-Begriff und darauf aufbauend in Abschnitt 2.2.2 mit der Modellierung. Die Bedeutung und das Verstehen von Modellen behandeln wir in Abschnitt 2.2.3. In Softwareentwicklungsprojekten arbeiten typischerweise mehrere Personen, Modelle werden in einem Softwareentwicklungsprojekt häufig von verschiedenen Personen verwendet und auch erstellt bzw. integriert, wir beschäftigen uns mit der gemeinsamen Modellierung in Abschnitt 2.2.4. Modelle werden im Softwareentwicklungsprozess in verschiedenen Phasen auf unterschiedliche Weise verwendet, wir beschäftigen uns mit der Rolle von Modellen in der Softwareentwicklung in Abschnitt 2.2.5.

Ontologien sind in dieser Arbeit ein wichtiges Mittel zur Beschreibung eines gemeinsamen Verständnisses über die Problem- und Lösungsdomäne. Wir führen in Abschnitt 2.3 in alle wichtigen Grundlagen rund um Ontologien ein. Wir definieren den in vielen wissenschaftlichen Disziplinen verwendeten Ontologie-Begriff in Abschnitt 2.3.1 gehen in Abschnitt 2.3.2 auf die Motivation ein, warum Ontologien in der Informatik eingesetzt werden. Es gibt eine Vielzahl von Arten von Ontologien, wir geben in Abschnitt 2.3.3 einen Überblick über diese. Den wichtigen Begriff des ontologischen Übereinkommens, definieren wir in Abschnitt 2.3.4. In Abschnitt 2.3.6 gehen wir auf die Entwicklung einer Ontologie ein. Der in dieser Arbeit vorgeschlagene Modellierungsansatz beruht zum Teil auf den Ansätzen, die in der Ontologie-Erstellung etabliert wurden. In Abschnitt 2.3.7 gehen wir auf Kriterien ein, die bei der Entwicklung einer Ontologie beachtet werden sollte. Diese Kriterien wirken sich unmittelbar auf die Lesbarkeit einer Ontologie aus und sind daher auch für diese Arbeit wichtig. Abschließend gehen wir in Abschnitt 2.3.8 darauf ein, wie Ontologien evaluiert werden können.

In Abschnitt 2.4 gehen wir auf verwandte Arbeiten ein.

2.1 Domänenwissen

In Abschnitt 2.1.1 wird der Begriff *Wissen* im Kontext der Informationstechnologie diskutiert, sowie die dessen Ausprägungen erläutert.

In Abschnitt 2.1.2 wird der in der Informatik häufig verwendete Begriff *Domäne* eingegrenzt. Darauf aufbauend wird in Abschnitt 2.1 der Wissens-Begriff in Anbetracht der Verwendung innerhalb einer bestimmten Domäne präzisiert.

Die Rolle von Domänenwissen in der Systementwicklung wird in Abschnitt 2.1.4 diskutiert.

2.1.1 Wissen

Der Begriff Wissen stammt (je nach Quelle) aus dem althochdeutschen (*wizzan* – ich habe gesehen) bzw. dem Lateinischen (*videre* – sehen) [Wal38]. Der Begriff Wissen wird in der Informatik unterschiedlich verwendet. U.a. schreibt Steinmüller [Ste93],

dass Wissen die zweckorientierte Vernetzung von Information¹ ist – Wissen besteht aus vielen Informationen und ist eine Anhäufung ideeller Modelle.

Für Strohner [Str90] ist Wissen die Abbildung der Realität (Verhältnisse, Zustände und Vorgänge) auf Modelle, über die ein Individuum oder eine Organisation verfügt. Ähnlich argumentiert Hefke [Hef08], Wissen entstehe durch Interpretation, Verarbeitung und Vernetzung von Informationen durch das Bewusstsein.

Wissen kann nicht einfach von Mensch zu Mensch weitergereicht werden [e-t10], da Wissen bei der Auseinandersetzung eines Individuums mit seiner Umwelt entsteht [MGR95].

Vernetzte Informationen müssen derart aufeinander bezogen sein, dass sie kohärent und nachvollziehbar stimmig sind, weiters ist Wissen durch individuelle Erfahrungen geprägt und an Personen gebunden. Ebenso wird Wissen auch in unterschiedlichen Kontexten interpretiert [Hef08].

Steinmüller stellt in [Ste93] einige Schlussfolgerungen über Wissen auf (gekürzt):

- ▶ Wissen existiert immer nur als Wissen über etwas/jemanden (eine modellierte Wirklichkeit = das Wissensoriginal), das also auch pragmatisch oder stigmatisch² richtig oder falsch abgebildet sein kann.
- ▶ Es ist immer nur Wissen jemandes (das ein Wissenssubjekt ausgewählt und transformiert hat – Strukturreduktion und -amplifikation), so dass Wissen über beliebige Objekte immer reduziert, also prinzipiell unvollständig ist.
- ▶ Es ist stets Wissen für einen Zweck (das ein Wissenssubjekt für einen Zweck geformt hat) – die Wissensfunktion also, die man kennen muss, um die Validität und Qualität dieses Wissens für meine oder andere Zwecke beurteilen zu können. Wissen ist zweckrelativ. Außerhalb des Zweckbereichs ist Wissen normalerweise unbrauchbar, in der sozialen Folge sozialschädlich (Verbot der Zweckentfremdung). Darüber kann auch eine objektivierte, d.h. vom Subjekt losgelöste, Erscheinungsform – etwa bei Expertensystemen – nicht hinwegtäuschen.
- ▶ Wissen hat also nur einen Wert für jemanden und somit keinen absoluten Wert; ist gleichwohl nicht wertfrei, sondern wertvoll, wenn man weiß, was wann wofür und für wen von welchem Wert ist.
- ▶ Wissen ist kontextabhängig, subjektiv- bzw. aspekt-objektiv und setzt die Kenntnis seiner Herkunft voraus.

Zusammenfassend bezeichnet nach Probst et al. [PRR06] Wissen, die Gesamtheit der Kenntnisse, die Individuen zur Lösung von Problemen einsetzen. Weiters führen Probst et al. [PRR06] an, dass dies sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen umfasst. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebun-

¹ Informationen sind Daten, die in einem Bedeutungskontext stehen [Nor05] – wobei Daten zum Zweck der Verarbeitung zusammengefasste Zeichen (Buchstaben, Ziffern, Sonderzeichen) die einer Ordnungsregel genügen [Nor05] und die aufgrund bekannter oder unterstellter Abmachungen Informationen (d.h. Angaben über Sachverhalte und Vorgänge) darstellen [WLS12] sind.

² Stigmatisch – in diesem Zusammenhang – deutet auf eine Abbildung mit geringer optischen Abweichung hin.

den – es wird von Individuen konstruiert und repräsentiert und deren Erwartungen über Ursache-Wirkungs-Zusammenhänge, so Probst et al. [PRR06] weiter.

Wissen ist zweckrelativ, kontext- und aspektabhängig, verhaltensrelevant und deshalb nie in einem schlichten Sinn objektiv [Ste93]. Das bedeutet, dass bei der Verarbeitung von Wissen immer alle Rahmenbedingungen betrachtet werden müssen, um Interpretationsfehler zu minimieren [RK96].

Die Betrachtung von Wissen in dieser Arbeit. Wir möchten die Aussage von Rehäuser und Krcmar [RK96] herausheben, wonach bei der Verarbeitung von Wissen immer alle Rahmenbedingungen betrachtet werden müssen, um Interpretationsfehler zu minimieren. Dieser Sachverhalt ist für diese Arbeit ein wichtiger Antrieb. Wir betrachten das Gesamtverständnis eines Softwareprojektes, bestehend aus Problem- und Lösungsdomäne, als das Wissen, das notwendig ist um ein Softwareentwicklungsprojekt durchführen zu können. Wie Selic [Sel12] in der Praxis feststellte ist dieses Gesamtverständnis häufig nicht vorhanden. In dieser Arbeit versuchen wir auf einer hohen Abstraktionsebene ein solches Gesamtverständnis zu schaffen, das als Ausgangsbasis dient, um weitere Detaillierungen an Ausschnitten des Gesamtverständnisses vorzunehmen.

Wissen wird in der Literatur typischerweise kategorisiert, 1966 führte Polanyi in [Pol66] die Begriffe explizites und implizites Wissen erstmals ein, die im folgenden erläutert werden.

Implizites und explizites Wissen. Für Polanyi [Pol66] ist implizites Wissen, das nicht formalisierte Wissen einer Person. Implizites Wissen bezeichnet das Wissen, das nicht bewusst verfügbar ist, es wird auch stilles Wissen genannt. Wissen ist vielfach nicht explizit vorhanden. In der Softwareentwicklung kann Wissen über eine Domäne z. B. implizit im Programmcode vorhanden sein [Rat09].

Explizites Wissen bezeichnet das Wissen, über das ein Lebewesen bewusst (explizit) verfügt. Explizites Wissen kann ausgedrückt werden, z. B. in Form von Daten, Formeln, Spezifikationen, Anleitungen, usw.

Implizites Wissen ist etwas Persönliches und ein schwer zu formulierender Begriff. Implizites Wissen beinhaltet subjektive Einsichten, Intuitionen und Ahnungen. Implizites Wissen ist tief verwurzelt in Tätigkeiten, Abläufen, Einsichten und Ideale, Werte und Emotionen, usw. [NTK00, Pol66]. Implizites Wissen hat nur für den Wissensträger die eine für einen selbst wahre Bedeutung innerhalb des persönlichen Deutungsrahmens³.

In Abbildung 2.2 eingezeichnet ist auf der Y-Achse die epistemologische Dimension (die Erkenntnis, bzw. explizites Verfügbar sein von Wissen) und auf der X-Achse die ontologische Dimension des Wissens nach Nonaka et al. [NT95]. Wir gehen in dieser Arbeit davon aus, dass sich implizites Wissen nur in Teilen explizieren lässt.

³ Der Deutungsrahmen, u.a. identifiziert und beschrieben von Entman [Ent93], bezeichnet die in einem Menschen bereits vorhandene, von externen Einflüssen (gesellschaftlich, individuell) geprägte Wissensstruktur [AS03], die dazu dient, einer aufgenommenen Informationen eine (persönliche) Bedeutung, durch Anreicherung bzw. Interpretation durch bereits vorhandener Wissensstrukturen, zu geben. Die Definition befindet sich auf Seite 28.

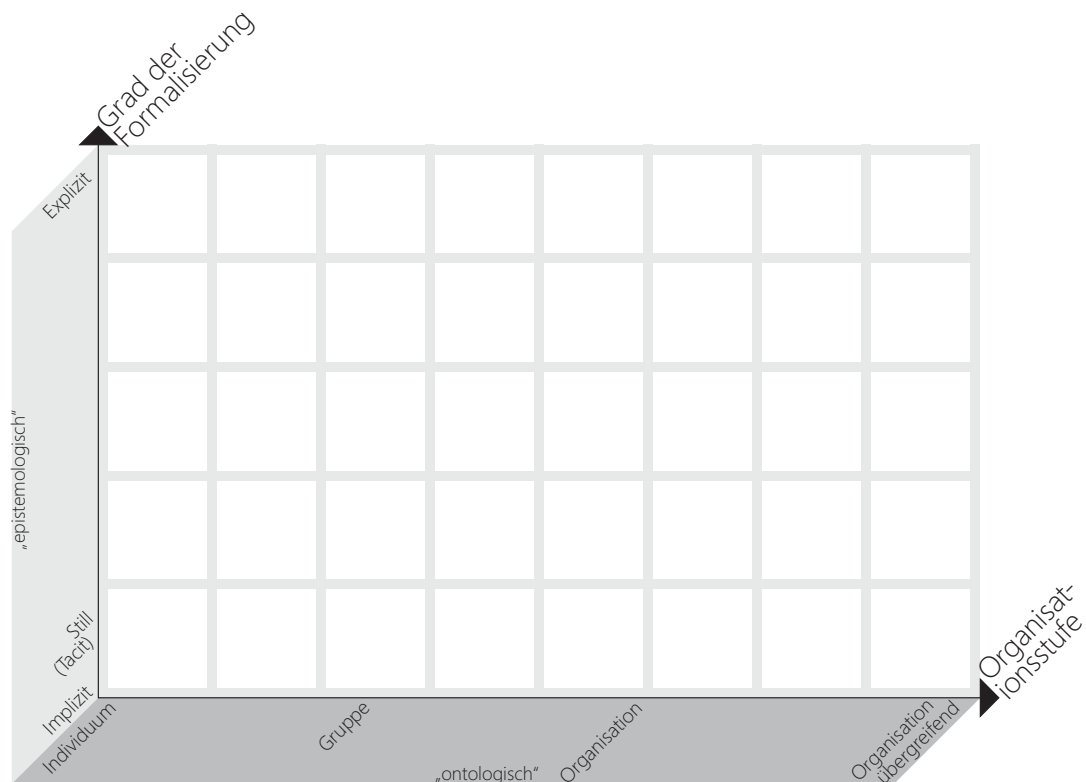


Abbildung 2.2: Die Dimensionen des Wissens in Unternehmen, angelehnt an [NT95]

Das von Nonaka et al. [NTK00] eingeführt SECI-Modell unterliegt dem SECI-Prozess, der Prozess ist ein iterativer Prozess und besteht aus vier sogenannte Modi (Sozialisierung, Externalisierung, Kombination und Internalisierung). Zusätzlich wird zwischen stillem Wissen (orig. *tacit knowledge*) und explizitem Wissen unterschieden. Nonaka et al. [NTK00] nennt die Interaktion zwischen diesen beiden Arten von Wissen Wissenskonversation. Diese Wissenskonversation führt (nach Nonaka et al. [NTK00]) zu einem qualitativ und quantitativ besser verfügbarem stillem und explizitem Wissen. Diese Wissenskonversationen sind in den jeweiligen Ecken in Abbildung 2.3 eingezeichnet.

Diese Modi sind [NTK00, NTM12]:

- ▶ **Sozialisierung:** Bezeichnet den Prozess stilles Wissen durch geteilte Erfahrungen zu festigen. Stilles Wissen kann nur über gemeinsame Erfahrungen (z. B. in einer Gruppe) geteilt werden.
- ▶ **Externalisierung:** Bezeichnet den Prozess das stille Wissen zu explizieren. Es kann danach mit anderen geteilt werden. Die Externalisierung stellt dabei eine besondere Herausforderung dar, Nonaka et al. [NT95] stellt dazu Extraktionstechniken vor.
- ▶ **Kombination:** Bezeichnet den Prozess explizites Wissen zu einem aussagekräftigeren expliziten Wissen zusammen zu setzen. Dabei wird expliziertes Wissen innerhalb oder außerhalb eines Betrachtungsumfeldes verwendet, um das explizierte Wissen aussagekräftiger und es ggf. zu neuem explizitem Wissen zu machen.

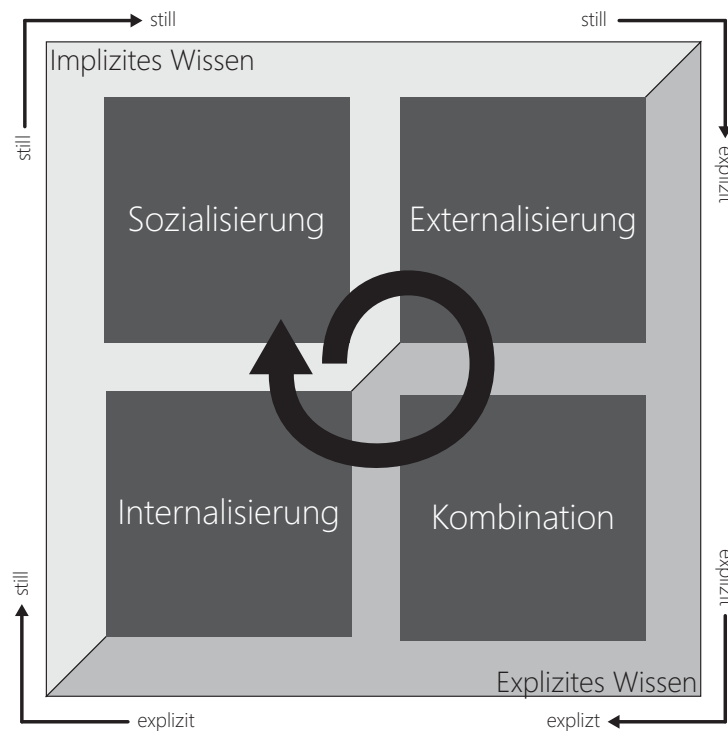


Abbildung 2.3: SECI-Modell mit eingezeichneten Wissenskonversationen und Prozessspirale, angelehnt an [NTK00]

- **Internalisierung:** Bezeichnet den Prozess das explizite Wissen zu verinnerlichen, damit es im Alltag wieder Anwendung findet. Dieser Prozess ist stark verbunden mit der Kompetenzbildung (siehe nachfolgender Abschnitt).

Auch explizites Wissen ist für eine Person erst nach Interpretation weiterverwertbares Wissen. Steinmüller stellt dazu fest [Ste93]: „Wissen entsteht ausschließlich durch menschliche Interpretation“.

Die Betrachtung von explizitem und implizitem Wissen in dieser Arbeit. In dieser Arbeit versuchen wir die Erkenntnisse von Nonaka et al. [NT95] in frühen Phasen des Requirements Engineerings anzuwenden. Sozialisierung stellt jenen Schritt in der Wissenskonversation dar, in dem eine Gruppe eine gemeinsame *Erfahrung* teilt. In dieser Arbeit versuchen wir dies mit der gemeinsamen Modellierung der Problem- und Lösungsdomäne zu erreichen. Die Externalisierung geschieht mit der erarbeiteten Vorgehensweise, die in ihren Einzelaktivitäten aus einer Vielzahl von Fragen besteht. Dies ist der von Nonaka et al. [NT95] angesprochene Extraktionsvorgang, der im Rahmen der Externalisierung stattfindet. Die Kombination findet an zwei Stellen im Ansatz statt, einerseits wird versucht, das neu hinzugekommene extrahierte Wissen innerhalb einer Iteration dem bereits vorhandenem Wissen zuzuordnen bzw. dessen Erkenntnisse zu integrieren und andererseits wird die Kombination durchgeführt, indem Anknüpfungspunkte zu weiteren Modellierungstechniken expliziert werden. Die Internalisierung ist das, was innerhalb im Projekt stattfindet. Das explizierte Wissen wird im Entwicklungsprozess eingesetzt. Erst wenn das Wissen ein-

setzbar ist, findet nach dem SECI-Modell die Internalisierung statt. Dafür muss das explizierte Wissen so repräsentiert werden, dass es verstanden werden kann.

Wissensmanagement. Wissen muss, um für ein Unternehmen bzw. einem Zweck verfügbar zu sein, auch verwaltet werden. Dieser Abschnitt führt kurz in das Wissensmanagement ein. Wissen über etwas zu explizieren, stellt für Unternehmen einen Mehrwert dar, sofern es zu einem geeigneten Zeitpunkt auch in geeigneter Form zur Verfügung steht. Wissensmanagement muss den in den Abschnitten über das Wissen eingeführten Problemen Antworten bieten. Dies inkludiert z. B. das Explizieren von Wissen, das nur implizit in einem Mitarbeiter verfügbar ist.

Stilles bzw. implizites Wissen ist auch im Requirements Engineering ein großes Problem [NE00]. Dieses Wissen für verschiedene unternehmensweite Aufgaben zu verwalten, damit es zum Zeitpunkt, an dem es benötigt wird, verfügbar ist, ist eine der Aufgaben des Wissensmanagements. Wissen wird immer für einen bestimmten Zweck erhoben. Wissensmanagement sollte immer einem konkreten Ziel folgen und daher pragmatisch betrieben werden. Wissen sollte vor allem einfach dargestellt und darauf aufbauend, auch einfach nutzbar sein [PRR06].

Ziele des Wissensmanagements sind nach Hefke [Hef08] u.a.: Überführung von Unternehmensproblemen in Wissensprobleme, Beurteilungsmöglichkeit der Wirkung von Entscheidungen auf organisationale Wissensbestände, Vermeidung von Pauschallösungen und Hilfe beim Verständnis wissensspezifischer Probleme, Problemorientiertheit, Bereitstellung eines handlungsorientierten Analyserasters, Kriterien zur Messbarkeit des Erfolgs, Integration bestehender Lösungsansätze und Berücksichtigung bereits existierender Systeme, Verständlichkeit und Vermittelbarkeit im Alltag.

Wilkesmann und Würmseer [WW07] führen an, dass Wissen in einem Kontext von Relevanz integriert werden muss. Dies beruht auf der Ausführung, dass Wissen aufbauend auf vorhandenem Wissen ist und dies die Unterscheidung zur Information ausmacht. Für das Wissensmanagement bedeutet dies, dass das Nutzbarmachen des Wissens besonders von der Relevanz des Wissens abhängt. Die Relevanz ist von den Zielen abhängig. Probst et al. [PRR06] hat dazu die Kernprozesse des Wissensmanagements formuliert, um die Zusammenhänge und Aufgaben zu explizieren, was in Abbildung 2.4 dargestellt ist.

Die folgenden Beschreibung zu den Kernprozessen stammen u.a. aus [PRR06, e-t10, Hef08]:

- ▶ **Erwerb und Akquisition:** Unternehmen importieren einen erheblichen Teil ihres Wissensbedarfs aus externen Quellen. Durch Rekrutierung von Experten oder die Akquisition von Unternehmen können Firmen sich Know-How einkaufen, das sie nicht aus eigener Kraft entwickeln können.

Fragestellung: Welche Fähigkeiten kauft sich das Unternehmen extern ein?

- ▶ **Konstruktion und Entwicklung:** Die Konstruktion und Entwicklung umfasst alle Anstrengungen, mit denen das Unternehmen sich bewusst um die Produktion bisher intern noch nicht bestehender oder gar um die Schaffung intern und extern noch nicht existierender Fähigkeiten bemüht.

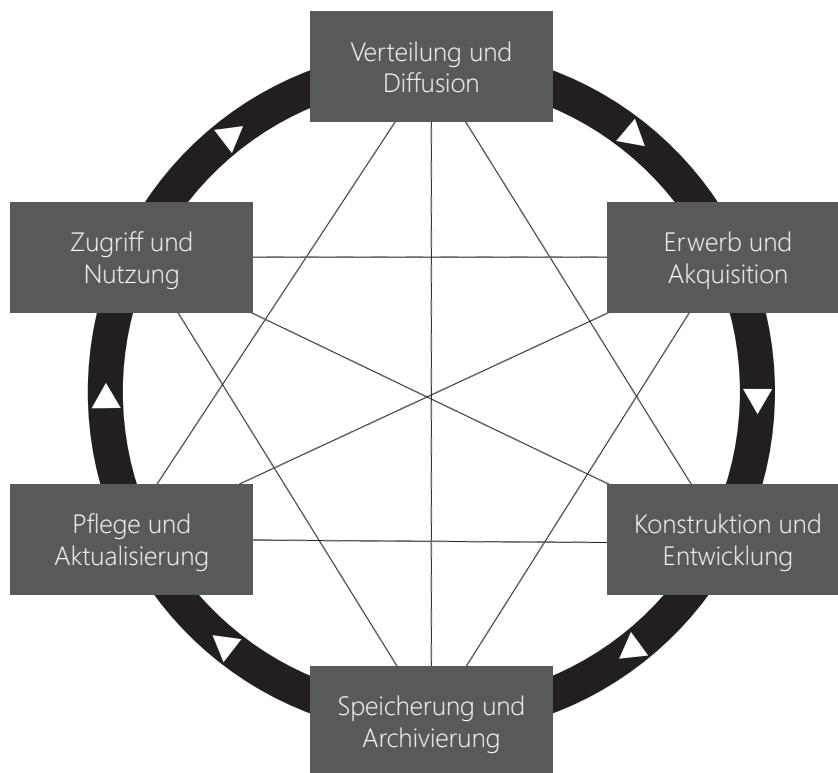


Abbildung 2.4: Kernprozesse des Wissensmanagements, angelehnt an [PRR06, e-t10]

Fragestellung: Wie baut das Unternehmen neues Wissen auf?

- ▶ **Speicherung und Archivierung:** Erworbene Fähigkeiten stehen nicht automatisch für die Zukunft zur Verfügung. Der Prozess der Bewahrung in Form von Speicherung und Archivierung beruht auf der effizienten Nutzung verschiedenster organisationaler Speichermedien für Wissen.

Fragestellung: Wie schützt das Unternehmen sich vor Wissensverlust?

- ▶ **Pflege und Aktualisierung:** Unternehmen müssen das Wissen nach Konstruktion, Entwicklung, Erwerb oder Akquisition aktuell halten.

Fragestellung: Wie kann ein aktives Pflege- und Aktualisierungsumfeld geschaffen werden?

- ▶ **Zugriff und Nutzung:** Der Zugriff und die Nutzung ist der produktive Einsatz organisationales Wissen zum Nutzen des Unternehmens.

Fragestellung: Wie stellt das Unternehmen die Anwendung des Wissens sicher?

- ▶ **Verteilung und Diffusion:** Die Teilung und Verteilung von Wissen ist eine zwingende Voraussetzung um isoliert vorhandene Informationen oder Erfahrungen für das Unternehmen nutzbar zu machen.

Fragestellung: Wie und wo wird das Wissen benötigt und wie wird es richtig verteilt?

Die Kernprozesse des Wissensmanagements liefern einen guten Start für die Prozes-

se, welche im Umgang mit Wissen involviert sind. Wichtig ist in diesem Zusammenhang das Herausarbeiten eines Wissensziel, beim Umgang mit Wissen. Das Wissensziel legt fest, auf welchen Ebenen eines Unternehmens welche Fähigkeiten aufgebaut werden sollen [PRR06]. Das Wissensziel wirkt folglich direkt auf die Wissensidentifikation, diese bildet die Basis für Maßnahmen und die Festlegung des Wissens, welches im Rahmen des Wissensmanagements betrachtet werden soll.

Im Umgang mit Wissen, unterschiedslos um welche Art von Wissen es sich handelt, spielt der Managementprozess und alle damit involvierten Aktivitäten eine wichtige Rolle. Dies gilt auch im Umgang mit Wissen in dieser Arbeit. Viele der aus dem Wissensmanagement bekannten Techniken (z. B. zur Akquise) finden sich im Ansatz – in angepasster Form – wieder.

Die Rolle vom Wissensmanagement in dieser Arbeit. Wissensmanagement muss pragmatisch betrieben werden [PRR06]. Ausgehend vom Wissensziel, dem Gesamtverständnis von Problem- und Lösungsdomäne innerhalb der relevanten Stakeholder innerhalb eines Softwareentwicklungsprojektes, behandeln wir die einzelnen Schritte im Wissensmanagementprozess in dieser Arbeit, wie folgt, um die Nutzbarmachung von Wissen zu ermöglichen:

- (a) Erwerb und Akquisition: Dieser Schritt im Softwareentwicklungsprozess ist die Befragung von relevanten, identifizierten Stakeholdern, die dann als Domänenexperten herangezogen werden. Dies wird mit der Methode, vorgestellt in dieser Arbeit, unterstützt.
- (b) Konstruktion und Entwicklung: Die Integration eines Softwareproduktes in eine unternehmerische Tätigkeit ist die Konstruktion und Entwicklung. Das Wissen, wie Softwareprodukte in diese unternehmerische Tätigkeit integriert sein werden, ist Teil der Konstruktion und Entwicklung. Die Feststellung, welches Wissen bei der Integration relevant ist, wird von der Methode in dieser Arbeit unterstützt.
- (c) Speicherung und Archivierung: Dies wird in dieser Arbeit nicht behandelt. Typischerweise werden Softwareprojekte mit Unterstützung von Versionsverwaltungssystemen durchgeführt. Diese eignen sich zur Speicherung und Archivierung der erstellten Artefakte.
- (d) Pflege und Aktualisierung: Dies ist Teilbereich der Methode. Gerade in frühen Phasen des Softwareentwicklungsprozesses gibt es häufig Änderungen in Artefakten. Der in dieser Arbeit vorgestellte Modellierungsansatz strebt die Explizierung eines Gesamtverständnisses an, auf dem andere Artefakte basieren. Die Aktualität und Konsistenz dieses ist besonders wichtig. Dies wird durch Konsistenzregeln und dem Syntax, in dem das Gesamtverständnis erfasst wird, gesichert.
- (e) Zugriff und Nutzung: Die relevanten Stakeholder sind an der Erstellung des Gesamtverständnisses beteiligt. Dieses dient als Ausgangsbasis für weitere Artefakte. Die Sicherstellung der Verwendung des Gesamtverständnisses erfolgt durch die vorgegebene Methodik.
- (f) Verteilung und Diffusion: Die richtigen Stakeholder müssen Zugriff auf dieses Gesamtverständnis und alle darauf aufbauenden Artefakte haben. Dies liegt in

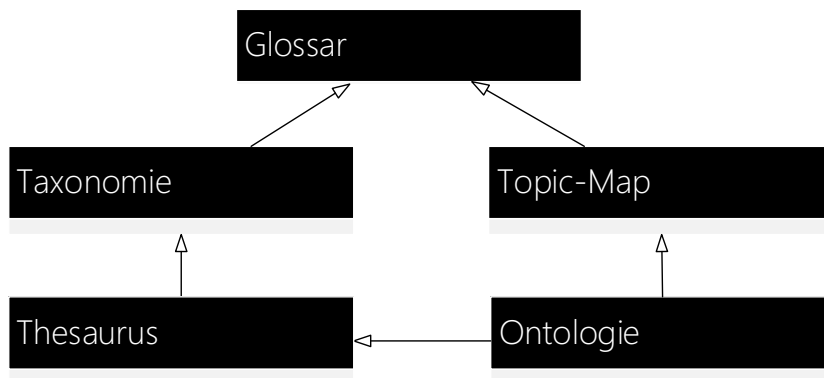


Abbildung 2.5: Die wichtigsten Vertreter der Wissensrepräsentationstechniken mit ihrem Zusammenhang

der Verantwortung des Entwicklungsteams.

Wissensrepräsentation. Wissen muss, um nicht nur mittels Sprache kommunizierbar zu sein, repräsentiert werden. Für diesen Zweck haben sich eine Menge von Wissensrepräsentationstechniken etabliert. Abbildung 2.5 zeigt die wichtigsten Vertreter (nach ihrer Anwendungshäufigkeit bzw. Beachtung, u.a. in [MB05]) der Wissensrepräsentationstechniken in ihrem Zusammenhang. Der Zusammenhang wird in der Abbildung mittels der Spezialisierungs-Beziehung ausgedrückt.

- ▶ **Glossar:** Ein Glossar enthält eine Liste von Begriffen und deren Erklärung.
- ▶ **Taxonomie:** Eine Taxonomie wird verwendet, um Begriffe einer bestimmten Domäne, Kategorien bzw. Klassen unterzuordnen. Taxonomien erleichtern den Umgang mit Einzelfällen da sie Aussagen über eine ganze Kategorie bzw. Klasse treffen und sind deshalb für den Menschen sehr wichtig. Kategorien fördern das Verständnis, da sie den Menschen dazu befähigen, sich der (allgemeinen) Bedeutung eines Begriffes im Klaren zu sein, dies führt zu einem besseren Verständnis [FK03].
- ▶ **Thesaurus:** Ist ein kontrolliertes Vokabular, dessen Begriffe durch Relationen miteinander verbunden sind. Ein Thesaurus versucht eine Domäne genau zu beschreiben und besteht aus systematisch geordneten Begriffen, die in einer thematischen Beziehung zueinander stehen [AC04]. In einem Thesaurus werden vorwiegend Begriffe gesammelt, die über Synonym, Ober- bzw. Unterbegriff in Beziehung stehen [Gar04]. Ein Thesaurus geht über die reine Klassifizierung einer Taxonomie hinaus.
- ▶ **Topic Map:** Eine Topic-Map ist ein Begriffsnetz, das von mehreren zentrierten Begriffen ausgeht [Gar02, Gar04] (im Gegensatz zu Mind-Maps, die nur von einem wichtigen Begriff ausgehen). Bei Topic-Maps ergibt sich die Semantik aus der Strukturierung der Begriffe. Topic-Maps sollen dem Menschen dabei helfen, über die Beziehungen und Begriffe in der Topic-Maps reflektierend und analysierend nachzudenken und anzupassen.
- ▶ **Ontologie:** Ontologien sind die höchste Formalisierungsstufe der in der Wis-

sensrepräsentation typischerweise angewendeten Repräsentationstechniken. Siehe dazu Abschnitt 2.3.

Wissensrepräsentation in dieser Arbeit. Da, wie in Abschnitt Wissensmanagement erwähnt, die Konsistenz und Sicherung der äußeren Form des repräsentierten Wissens eine besondere Rolle in dieser Arbeit spielt, haben wir uns entschlossen, Ontologien als Repräsentationsform zu verwenden. Diese werden in Abschnitt 2.3 im Detail erläutert.

2.1.2 Domäne

[Bjø12] sieht im Rahmen einer Domäne ein von menschlichen Sinnen wahrnehmbares Phänomen betroffen. In seinem Sinne ist eine Domäne ein Bereich menschlicher Aktivität der durch wahrnehmbare Phänomene charakterisiert wird.

Tracz bezeichnet in [Tra94] eine Domäne als eine Menge von gemeinsamen Problemen oder Funktionen die eine Anwendung in dieser Domäne lösen kann. Spezifischer für die Softwareentwicklung aber konform zur Definition von Tracz, bezeichnet Prieto-Díaz in [PD90] eine Domäne im Kontext der Softwareentwicklung als einen Bereich, für den eine Anwendung erstellt wird bzw. wurde.

Unter Domäne kann auch verstanden werden (angelehnt an die Aufteilung in [Sch00]):

- ▶ der Geschäftsbereich,
- ▶ eine Sammlung von Problemen,
- ▶ die Sammlung von Lösungen und Realitäten sowie
- ▶ ein Wissensbereich mit einer gemeinsamen Terminologie

Wir verwenden den Begriff Domäne in dieser Arbeit als einen Anwendungsbereich, für den ein Softwaresystem entwickelt wird oder in dem ein Softwaresystem eingebettet ist.

Grenzen einer Domäne. Eine Domäne, als festgelegter Bereich über etwas, impliziert, dass eine Domäne abgrenzbar ist. In der Realität gestaltet sich diese Abgrenzung schwierig.

Nach Schmid [Sch00] können die Grenzen einer Domäne festgelegt werden, indem definiert wird, (a) was innerhalb einer Domäne liegt, (b) wo die Grenzen der Domäne liegen oder (c) was außerhalb der Domäne liegt.

Alle erwähnten Festlegungen haben das gleiche Problem, die Definition des Bereiches bzw. der Grenzen ist zumeist nur vage möglich, z. B. die Aussage: Alle Geschäftsprozesse, die für die Anwendung relevant sind. Zunächst klingt diese Aussage wie eine konkrete Festlegung, die auch nachvollziehbar und nachprüfbar ist. Durchaus können nun die Grenzen eines Unternehmens überschritten werden, weil auch Geschäftsprozesse relevant werden, die nicht zum Unternehmen gehören. Zusätzlich werden relevante Prozesse häufig in der Softwareentwicklung erst während

der Entwicklung identifiziert. Die Grenzen der Domäne sind daher sehr wohl klar definiert, aber die Menge der Prozesse innerhalb der Domäne wächst.

Diese und ähnliche Szenarien können für alle diese Festlegungen der Bereichsfindung gefunden werden.

Lam et al. [LM97] stellten fest, dass die Grenzen einer Domäne nicht einfach zu finden sind. In Abbildung 2.6 finden sich die möglichen Abgrenzungen, die Domänen ausgesetzt sind.

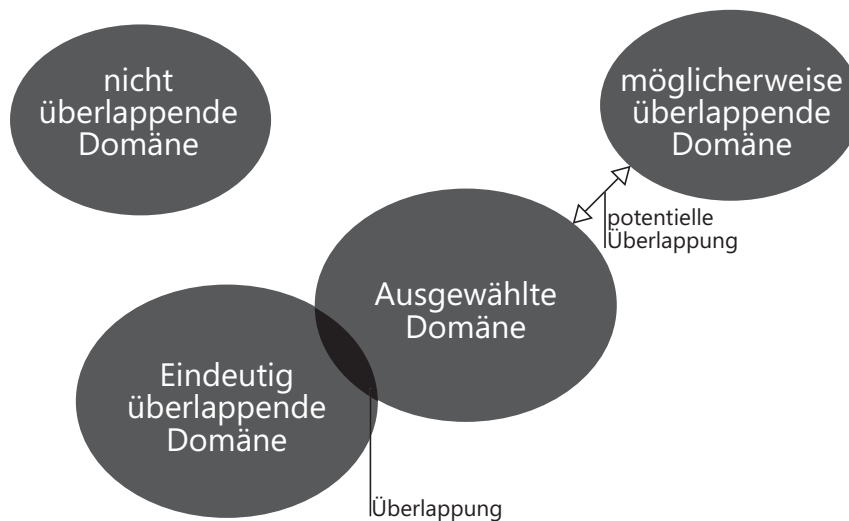


Abbildung 2.6: Mögliche Domänen-Überlappungen, angelehnt an die Darstellung in [LM97]

Lam et al. [LM97] stellten dazu weiter fest, dass Domänen sich überlappen können und diese Überlappung nicht immer eindeutig identifizierbar ist. Nicht überlappende Domänen sind Domänen, die klar abgrenzbar sind, bzw. in ihrem Inhalt nicht dem betrachteten Anwendungsbereich entsprechen und so eine Überlappung daher weitgehend ausgeschlossen werden kann. Domänen können sich durchaus eindeutig überlappen, wie es bei verschiedenen Domänen (Bereichen, z. B. in Form von Prozessen) eines Unternehmens der Fall ist, die ineinander über gehen. Zusätzlich gibt es die Möglichkeit, dass sich Domänen potentiell überlappen, z. B. im Falle der Festlegung der Domäne noch nicht sicher gesagt werden kann, dass sich Domänen überschneiden werden.

Problem- und Lösungsdomäne. Häufig wird in der Softwareentwicklung von Problem- und Lösungsdomäne gesprochen.

Abbildung 2.7 zeigt ein häufig auftretendes Problem, wenn über Problem- und Lösungsdomäne gesprochen wird. Für einige Softwareentwickler beginnt die Lösungsdomäne bei Artefakten, die Teil des umgesetzten Produktes sind, daher mit der Architektur- bzw. Grobentwurfsphase. Bei einige anderen, wie z. B. Hull [HJD11], beginnt die Lösungsdomäne bereits im Requirements Engineering. Anforderungen beschreiben naturgemäß eben Anforderungen an eine Lösung und sind, im Gegensatz zu den Zielen und der Beschreibung des Problembereiches, daher bereits Teil der Lösungsdomäne.

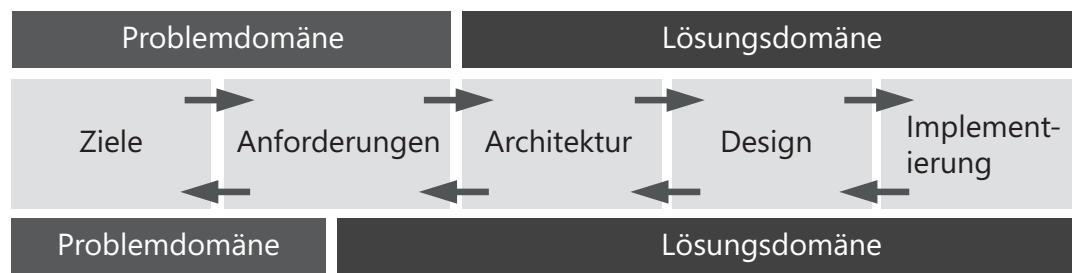


Abbildung 2.7: Abgrenzung Problem- und Lösungsdomäne

Die Unterscheidung wird auch aufgrund von Fragestellungen getroffen:

- ▶ Warum und was wird gebaut? Hier sprechen wir von der Problemdomäne. Häufig wird im Rahmen dieser Fragestellung auch von Stakeholder-Requirements gesprochen, da diese die Ziele, Wünsche usw. der Stakeholder repräsentieren [HJD11].
- ▶ Wie wird es gebaut? Hier sprechen wir von der Lösungsdomäne. Darunter können bereits die konkreten Anforderungen an ein Softwaresystem verstanden werden, die dann bereits zur Lösungsdomäne gehören.

Der Domänen-Begriff in dieser Arbeit. In dieser Arbeit verwenden wir die Begriffe wie folgt:

Wiederholung der Definition von Seite 3:

Definition Problemdomäne:

Unter Problemdomäne verstehen wir ein abgrenzbares Problemfeld oder den Anwendungsbereich für ein Softwaresystem.

Wiederholung der Definition von Seite 3:

Definition Lösungsdomäne:

Unter Lösungsdomäne verstehen wir einen abgrenzbaren Bereich, der sich nicht mit der Problemdomäne überschneidet und all das umfasst, was direkt zur Lösung eines Problems in der Problemdomäne beiträgt.

2.1.3 Domänenwissen

Iscoe [IWA91] formulierte folgende Merkmale von Domänenwissen: Es sei eher informell wie formal, eher implizit wie explizit, eher ad-hoc anstatt von allgemein verwendbar und nur unvollständig und indirekt über problemspezifische Sprachen erfassbar. Auch Jackson et. al ist in [JZ93] darauf eingegangen, dass Domänenwissen nicht immer formal erfassbar ist. Iscoe trifft keine Aussagen über die Art des Wissens, jedoch sehr wohl über den Zweck, den Domänenwissen erfüllen kann, z. B. im Requirements Engineering.

Wir unterscheiden persönliches Domänenwissen – daher, das Domänenwissen einer Person – und das geteilte Domänenwissen – daher das Domänenwissen einer Personengruppe.



Abbildung 2.8: Deutungsrahmen einer Person und dem darin eingebetteten Domänenwissen

Domänenwissen einer Person. Domänenwissen einer Person ist das Wissen, dass eine bestimmte Person über eine bestimmte Domäne im Deutungsrahmen der Person hat – initial ist es daher nicht expliziert und stilles Wissen, das mitunter auch schwer zu explizierendes Wissen sein kann. Das Domänenwissen einer Person hat nur innerhalb des Deutungsrahmens genau dieser Person die Bedeutung. Das gleiche Domänenwissen, im angenommenen Falle, dass das Wissen losgelöst vom Wissensträger übertragbar wäre, wird daher im Deutungsrahmen einer anderen Person eine abweichende Bedeutung haben. Abhängig von der Präzision der Modellierung, sowie dem ontologischen Übereinkommen über die verwendeten Modellierungsprimitive kann der Deutungsrahmen so beeinflusst werden, dass der Interpretationsraum verkleinert wird.

Definition Deutungsrahmen:

Der Deutungsrahmen, u.a. identifiziert und beschrieben von Entman [Ent93], bezeichnet die in einem Menschen bereits vorhandene, von externen Einflüssen (gesellschaftlich, individuell) geprägte Wissensstruktur [AS03], die dazu dient, einer aufgenommenen Informationen eine (persönliche) Bedeutung, durch Anreicherung bzw. Interpretation durch bereits vorhandener Wissensstrukturen, zu geben.

Abbildung 2.8 zeigt konzeptuell den Deutungsrahmen einer Person und das hervorgehobene Domänenwissen. Diese Abbildung symbolisiert, dass Domänenwissen in einer Person *eingebettet* ist und bei der Explizierung besonderen Wert auf den Deutungsrahmen von anderen Personen gelegt werden muss.

Domänenwissen einer Personengruppe. Das Domänenwissen einer Personengruppe zu kommunizieren (unabhängig davon, ob das Wissen expliziert wurde) führt zur bekannten Kommunikationslücke. Der Kommunikationsempfänger weist einen anderen Deutungsrahmen auf (z. B. weisen Begrifflichkeiten – aufgrund einer anderen Wissensstruktur – auf andere identifizierten Sachverhalte der Realität).

Abbildung 2.9 zeigt die überschneidenden Deutungsrahmen einer Personengruppe, bestehend aus drei Personen (P1, P2, P3). Bei der Kommunikation des Domänenwissens überschneiden sich die Deutungsrahmen, sofern sie von der gleichen Domäne *Wissen* haben und daher bereits Wissensstrukturen darüber aufweisen. Hier sei angemerkt, dass sehr wohl auch keine Überschneidung vorkommen kann, z. B. ein Kommunikationsempfänger nimmt einen Begriff auf (hört ihn, liest ihn, ...), dem dieser unbekannt ist, folglich ist eine Deutung für diesen Begriff nicht möglich. Bereits das Mitkommunizieren einer Kategorisierung des Begriffes könnte eine erfolgreiche

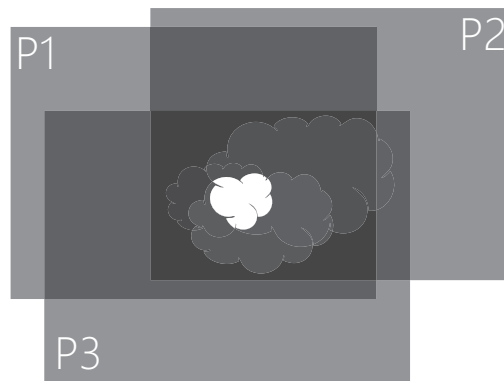


Abbildung 2.9: Deutungsrahmen und gemeinsames Domänenwissen

Deutung ermöglichen.

Kommunizieren nun Personen über eine bestimmte Domäne, variiert (a) das Wissen über die Domäne ebenso wie (b) der Deutungsrahmen. Das überschneidende Wissen, das auch eine möglichst gleiche Bedeutung aufweist ist im Bild in orange hervorgehoben. Die konzeptuelle Abbildung 2.9 zeigt das Problem der Kommunikation von Domänenwissen innerhalb einer Personengruppe auf. Ein wesentliches Problem ist es die Bedeutung der Begrifflichkeiten zu präzisieren und die Kommunizierbarkeit bestmöglich zu erhalten.

Der Vorgang der gemeinsamen Modellierung, daher das Erstellen eines Modell, das das gemeinsame Wissensoriginal abbildet, wird in Abschnitt 2.2.4 vorgestellt.

Die Betrachtung von Domänenwissen in dieser Arbeit. Domänenwissen spielt in dieser Arbeit eine wichtige Rolle. In einem Softwareprojekt kommt Domänenwissen auf sehr unterschiedliche Weise vor. Wir konzentrieren uns auf die Aspekte der Abläufe und des Aufbaus eines Unternehmens, die für das Wissen über die Integration eines Softwaresystems in seine Anwendungsdomäne relevant ist. Dies inkludiert in Teilen auch die Lösungsdomäne. Diese beiden Bereiche detaillieren wir, bis die Verknüpfung zu anderen Artefakt-Typen möglich ist. Wir erreichen so bereits in frühen Phasen des Requirements Engineerings ein Gesamtverständnis über die Problem- und Lösungsdomäne.

2.1.4 Arten von Wissen in der Softwareentwicklung

Hjørland et al. [HA95] bezeichnen Domänenwissen in der Softwareentwicklung als das Wissen über die Umgebung in dem ein System betrieben wird. Curtis et al. [CKI88] identifizierten die in Abbildung 2.10 dargestellten involvierten Domänen im Entwicklungsprozess.

Curtis et al. [CKI88] führten an, dass die meisten Personen, die im Entwicklungsprozess involviert waren, in einer oder zwei der angeführten Domänen Wissen hatten. Die Systemarchitekten hatten das relevante Übersichtswissen das für die Integration notwendig war.

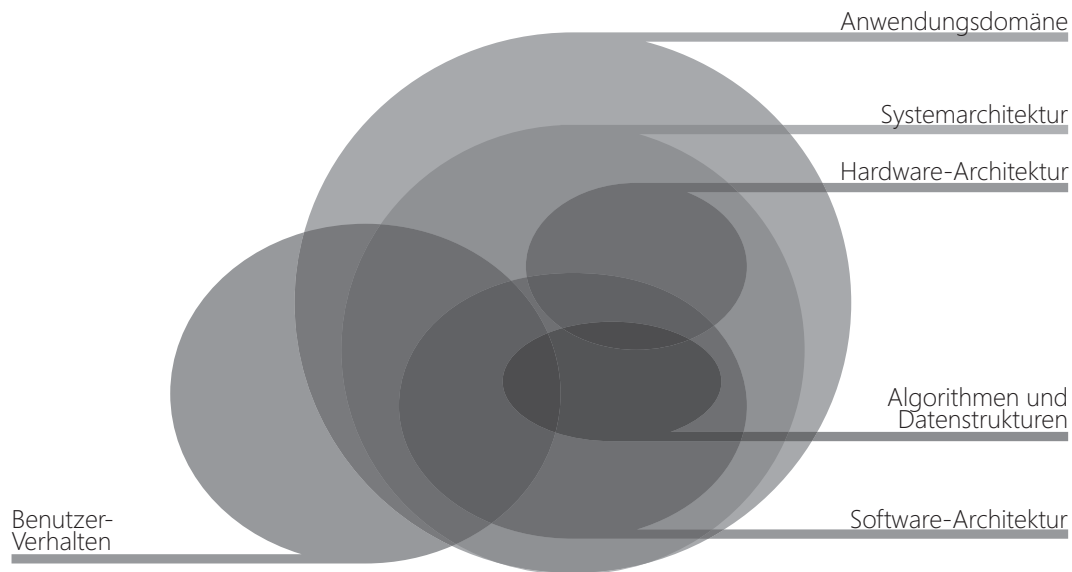


Abbildung 2.10: Domänenwissen nach den involvierten Teildomänen im Entwicklungsprozess, angelehnt an [CKI88]

Zur Bedeutung der Domänen hat ein Projektleiter in [CKI88] erwähnt, dass nicht das Entwickeln das Problem ist, sondern das Problem verstehen, ist das Problem. Dies unterstreicht die Wichtigkeit der Anwendungsdomäne im Entwicklungsprozess. Im Laufe eines Softwareentwicklungsprojektes müssen eine Vielzahl von Problemen gelöst werden. Abhängig von dem Wissen, das eine Person in den einzelnen Domänen hat, wird auch die Überwindung des Problems unterschiedlich schnell geschehen. Gerade Probleme, die außerhalb des Wissensbereiches von Entwicklern liegen bereiten einen nicht unerheblichen Aufwand. Das fehlende Wissen muss in Erfahrung gebracht werden, damit die Probleme überwunden werden können.

Die betrachteten Arten von Wissen in dieser Arbeit. Wie im Vorabschnitt erläutert, erheben wir im Rahmen der Ausführung der Vorgehensweise in dieser Arbeit vor allem Ausschnitte aus der Problem- und Lösungsdomäne. In der Abbildung 2.10 entspricht dies der Anwendungsdomäne und teilweise das Benutzerverhalten als Problem- und am ehesten die Software- bzw. Systemarchitektur als Lösungsdomäne.

2.2 Modellbildung

Modellbildung bezeichnet den Vorgang einen beschränkten Ausschnitt der Realität in ein für einen bestimmten Zweck geeignetes Modell zu überführen. Dies resultiert aus der Notwendigkeit heraus Information darzustellen, um sie einer Bearbeitung zugänglich machen zu können [BS03].

Dieser Abschnitt beschäftigt sich mit wichtigen Facetten des Modell-Begriffes in Abschnitt 2.2.1, geht in Abschnitt 2.2.3 auf die Semiotik von Modellen ein und überführt

dieses Wissen in den Begriff der Modellierung in Abschnitt 2.2.2. In dieser Arbeit spielt die gemeinsame Modellierung (die Modellierung eines Sachverhaltes in konsensuellen Modellen) eine wesentliche Rolle, dies wird in Abschnitt 2.2.4 beleuchtet. Abschnitt 2.2.5 führt generisch in die Bedeutung von Modellen im Softwareentwicklungsprozess ein.

2.2.1 Modell

Ein Modell ist nach Stachowiak [Sta73] eine Abbildung eines Originals, gekennzeichnet durch drei Merkmale. Diese drei Merkmale nach Stachowiak [Sta73], Hesse et. al [HM08] und Partsch [Par10] stellen wir wie folgt dar:

- ▶ **Abbildungs-Merkmal:** Jedes Modell steht für etwas anderes: sein Original [HM08], es gibt daher zu jedem Modell ein Original (vorher oder nachher), das selbst wieder ein Modell sein kann [Par10].
- ▶ **Reduktions-Merkmal:** Ein Modell weist nicht alle Eigenschaften des Originals auf, sondern nur einige – und auch diese möglicherweise in veränderter oder ähnlicher Form [HM08]. Partsch [Par10] führt dazu auf, dass nur relevante Informationen erfasst werden und auch neue hinzukommen können (in Form von modellspezifischen Eigenschaften).
- ▶ **Pragmatisches Merkmal:** Ein Modell hat den Zweck, unter bestimmten Bedingungen und bezüglich bestimmten Fragestellungen das Original zu ersetzen [HM08]. Die Zuordnung zwischen Modell und Original orientiert sich am Nützlichen und wird durch die Fragen, „Für wen?“, „Warum?“ und „Wozu?“ relativiert [Par10]. Ein Modell erlaubt somit die Beantwortung von Fragen in Hinblick auf das Original.

Der Modellbegriff im Modellierungsansatz in dieser Arbeit. Das Produkt der in dieser Arbeit propagierten Vorgehensweise ist ein Modell, das wir Basismodell nennen. Das Original unseres Modells ist ein unter mehreren Stakeholdern geteiltes Original, das vor allem im Deutungsrahmen der Stakeholder vorhanden ist. Dieses erheben wir. Unser Basismodell ist eine reduzierte Abbildung des Originals, das für einen ganz bestimmten Zweck erstellt wird. Der Zweck ist das Finden einer Gesamtvorstellung über Problem- und Lösungsdomäne zwischen den relevanten Stakeholdern für alle involvierten Stakeholder. Der Grund der Erstellung ist das bessere Überbrücken der Kommunikationslücke im Softwareentwicklungsprojekt um Interpretationslücken zu verkleinern.

2.2.2 Modellierung

In der Modellierung wird u.a. ein ausschnittsweises Abbild der realen Welt für eine Aufgabe der Informationsverarbeitung dargestellt und es werden Vorlagen für die Arbeitsweise eines informatischen Systems angegeben [BS03]. Wir fügen dem hinzu, dass die Informationsverarbeitung auch die Kognition meint, da Modelle in der Informatik auch der reinen Kommunikation zwischen Menschen dienen kann.

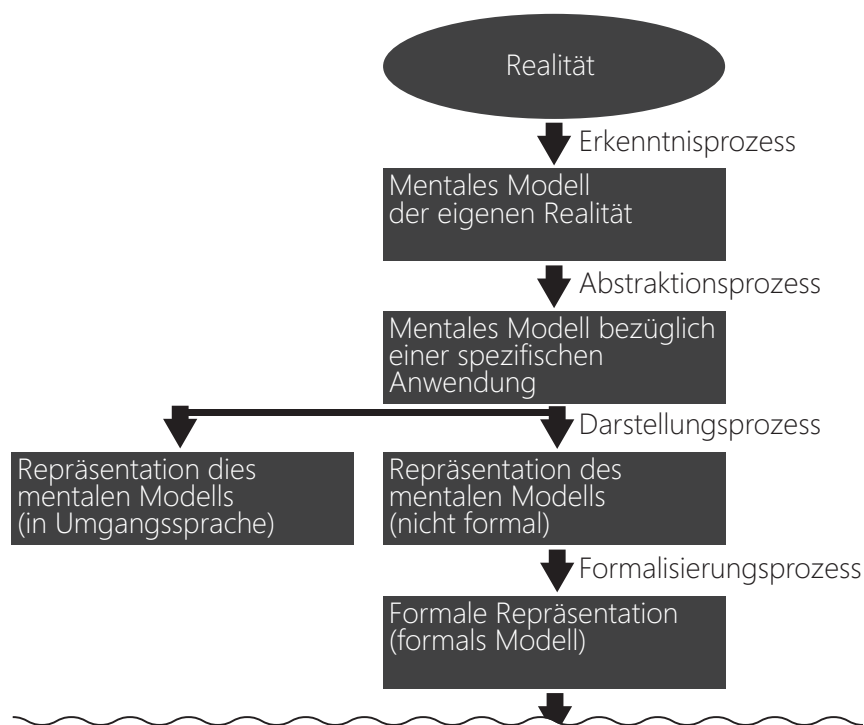


Abbildung 2.11: Abbildung der Realität in ein Modell

Abbildung 2.11 zeigt die Schritte und Zwischenergebnisse der Modellbildung, angelehnt an die Darstellung in [Par10].

Sobald die relevanten Gegenstände von ihrem Typ her identifiziert und auch voneinander unterscheidbar sind, lassen sich Beziehungen zwischen Gegenständen modellieren [BS03].

Broy und Steinbrüggen stellen in [BS03] fest, dass Experten unterschiedlicher Herkunft intensiv kommunizieren und kooperieren müssen, und zeigen vier Säulen auf, auf denen diese Kooperation und Kommunikation beruht (gekürzt):

- ▶ **Pragmatik:** Im Systems und Software Engineering hat sich eine Reihe von anschaulichen Beschreibungstechniken, insbesondere diagrammatischer Art, herausgebildet.
- ▶ **Formalisten:** Wie auch sonst im Ingenieursbereich sind die Aneignung und die Anwendung der einschlägigen Zweige der Mathematik unabdingbar. Nur dies gibt Gewähr für Präzision und Zuverlässigkeit, für Kostenbeherrschung und Nutzen der Produkte.
- ▶ **Methodik:** Abstraktion ist die wesentliche Technik, um die Komplexität von Software unter Kontrolle zu bringen. Sie spielt eine ähnliche vereinfachende Rolle wie Approximation in anderen Ingenieursbereichen.
- ▶ **Werkzeuge:** Methodisches Vorgehen erfordert im Allgemeinen den Einsatz von Spezialsoftware.

Vorgehensweise in der Modellbildung in dieser Arbeit. Die Vorgehensweise in dieser Arbeit unterstützt die Modellbildung durch eine Menge von Fragestellungen, die beantwortet werden sollen, um relevante Gegenstände zu identifizieren. Um die Unterscheidbarkeit zu unterstützen, muss jeder Gegenstand einem Stereotyp zugeordnet werden. Die Beziehungen zwischen den gefundenen Gegenständen müssen dann, wie die Gegenstände selbst im Modellierungsansatz identifiziert werden und werden, wie die Gegenstände, stereotypisiert, um die Unterscheidbarkeit zu unterstützen.

2.2.3 Modell-Semiotik

Semiotik ist die Lehre von den Zeichen und ihren Bedeutungen. Abbildung 2.12 zeigt das semiotische Dreieck von Ogden et al. [OR89], erweitert um ontologischen Aspekte durch Guarino et al. [GOS09] und Modell-Spezifika nach Hesse et al. [HM08] – manchmal auch in veränderter Form als Ullmann-Dreieck [Ull79] bezeichnet.

Zur allgemeinen Semiotik. Im Folgenden schildern wir einen Kommunikationsvorgang mit Hilfe von Zeichen⁴: Ein kommunizierendes Subjekt *K*, egal wie dessen ontologisches Übereinkommen zu einer Ontologie ist, findet sich im semiotischen Dreieck wieder. Der Sender einer Nachricht verwendet ein Zeichen, z. B. in Form eines Begriffes, dessen Bedeutung mental für ein Konzept steht. *K* verwendet das Zeichen um auf abstrakte oder reale Dinge der Realität zu referenzieren [GOS09]. Der Empfänger *E* der Nachricht, empfängt dieses Zeichen und ordnet es seiner konzeptuellen Bedeutung zu. *E* identifiziert durch die Interpretation (abhängig vom Kontext) ein Individuum der Realität, das (eventuell) gemeint war. Ein Problem tritt dann auf, wenn Nachrichten beim Empfänger eine andere Bedeutung haben, als sie der Sender initial gemeint hat.

Zum Modell-Verstehen. Abbildung 2.12 zeigt über die allgemein Semiotik hinaus noch die semiotischen Aspekte von Modellen nach Hesse et. al [HM08] und Falkenberg et. al [FHL⁺98]. Hesse et. al [HM08] sprechen davon, dass jedes Modell durch eine ausgewählte Modelldarstellung repräsentiert wird. Der gedankliche Kern eines Modelles liegt in seiner Semantik, der Modell-Konzeption, welche bestimmt, in welcher Weise das zugehörige Original aufgefasst und erklärt werden soll, welche seiner Eigenschaften weggelassen oder welche hinzugefügt werden sollen. Weiter führen Hesse et. al [HM08] aus, dass der pragmatische Bezug eines Modells (das Original) den Sinn und Zweck, Art und Tiefe der Modellbildung bestimmt. Wir sprechen beim Verstehen eines Modells vom Begreifen eines Sachverhaltes (in diesem Fall die Bedeutung einer Modell-Konzeption) in Abhängigkeit der persönlichen Modell-Interpretation. Kneer et. al [KS09] weist darauf hin, dass Verstehen über die bloße Kenntnisnahme hinaus geht und das intellektuelle Erfassen des Zusammenhangs inkludiert. Für das Verstehen eines Modells bedeutet dies, dass die Interpretation das Original bzw. Abbild für den Zweck identifizieren muss, für den das Modell erstellt wurde. Im Software-Prozess wird das Modell häufig als transientes Modell verwendet.

⁴ Ein Zeichen ist etwas festgelegtes, mit einer bestimmten Bedeutung verknüpfte und eine ganz bestimmte Information vermittelnde, grafische Einheit; auch Symbol [Dud11].

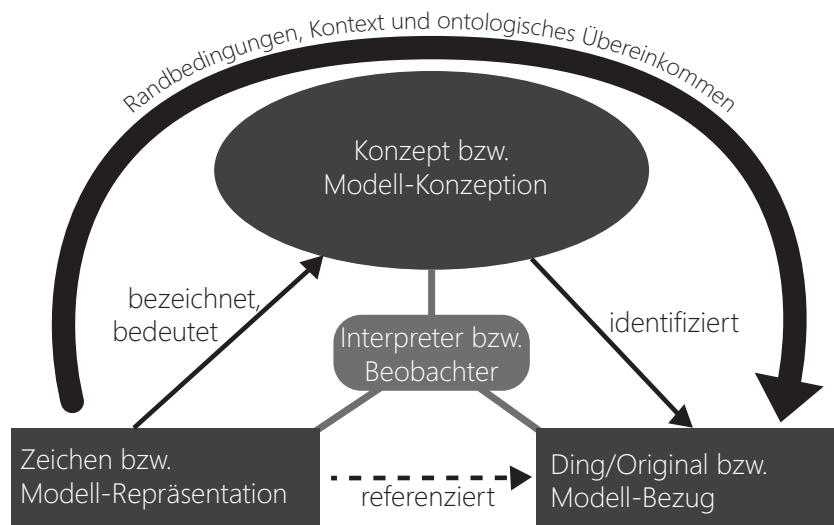


Abbildung 2.12: Semiotisches Dreieck nach [GOS09], erweitert um Modell-Spezifika und Interpreter bzw. Beobachter nach [FHL+98, HM08]

Hesse et. al [HM08] verweisen darauf, dass zwischen Repräsentation eines Modells und dem Gegenstand, den das Modell beschreibt, kein natürlicher Zusammenhang besteht, sondern erst durch den Modell-Interpret und seine persönliche Auffassung (Konzeption) dieser Zusammenhang entsteht. Der Erkenntnisprozess ist das Erkennen von Modellen der Realität durch (kognitive) Modelle [PLS73].

Modell-Semiotik in dieser Arbeit. In dieser Arbeit erstellen wir ein Modell, das als expliziertes Gesamtverständnis auf einer hohen Abstraktionsebene verstanden werden soll. Wie im Text erläutert, bestimmt der pragmatische Bezug die Tiefe der Modellbildung. Durch das Interpretieren des Modells, im Rahmens seines Zweckes, kann dessen Nutzen festgestellt werden. Wir haben in dieser Arbeit den Zweck des Modells eingegrenzt und versucht, in verschiedenen Softwareentwicklungsprojekten den Nutzen des Modells im Rahmen der Verwendung durch die Ersteller und Verwender bewerten zu lassen. Der Nutzen eines Modells zeigt sich erst durch die Verwendung desselbigen in jenem Kontext, für den das Modell vorgesehen ist.

2.2.4 Gemeinsame Modellierung

Thomas et al. [TT28] stellen fest, dass wenn ein Mensch eine Sache bzw. Situation als wahr annimmt, sie auch in jeder Konsequenz wahr ist, die daraus folgt. Dies führt dazu, dass eine falsche Vorstellung über die Welt mühsam angepasst werden muss, wenn sie falsch ist – und sich z. B. in einem nicht richtigen Domänenverständnis wieder spiegelt.

Das Verhalten eines Menschen u.a. in der Modellierung eines gemeinsamen Modells als gemeinsame Vorstellung der Realität, ist aufgrund der selektiven Wahrnehmung [Ven99] getrieben von den Erwartungen. Menschen konstruieren sich nicht nur ein Bild der Realität im Gedächtnis, sondern das Verhalten eines Menschen führt dazu, dass das konstruiert Bild der Realität im Gedächtnis auch Realität in der Umgebung

des Menschen wird [Ven99], dies wird in Abbildung 2.13 dargestellt, angelehnt an die Darstellung in [Ven99].

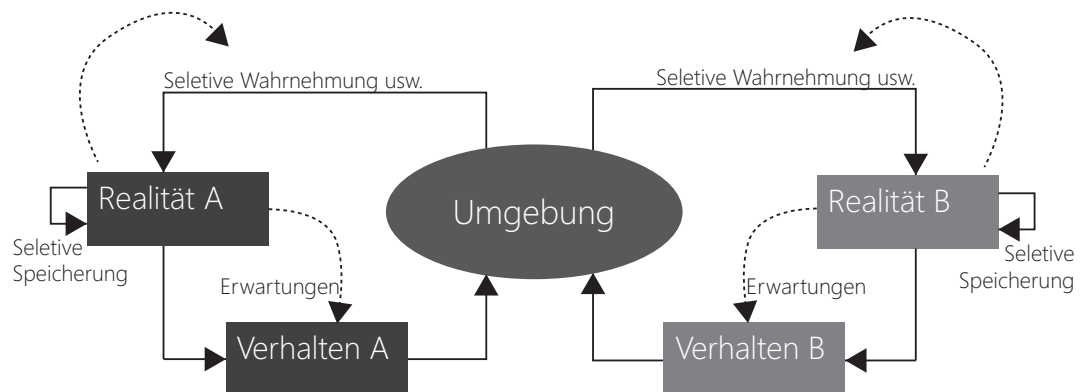


Abbildung 2.13: Wahrnehmung bzw. Deutungsrahmen zweier Personen einer gleichen Umgebung bzw. Realität, angelehnt an [Ven99]

Vennix [Ven99] weist besonders darauf hin, dass in der Findung eines Modells in der Modellbildung der Mensch alle Informationen ignoriert, die nicht mit seinem Weltbild zusammen hängen. Stattdessen suchen Menschen nach einer Erklärung für Information, die zu ihrem Weltbild passen und ebenso häufig wird so die mentale Abbildung der Realität angepasst, z. B. durch Erläuterung einer besonderen Situation, dass sie zum Weltbild passt [Ven99]. Menschen lernen daher sehr langsam neue Sachverhalte, sofern diese nicht zum eigenen Weltbild passen.

Gemeinsame Modellierung im Modellierungsansatz in dieser Arbeit. Eine Person kann daher nie das Wissen einer Domäne angebracht für eine Gruppe von Personen explizieren. Im Modellierungsprozess müssen verschiedenen Personen involviert sein, um die Anpassung des Originals an ein Weltbild einer einzigen Person zu verhindern und offene Diskussion zu fördern. Wir haben uns daher entschlossen und in verschiedenen Fallstudien getestet, von den im Vorgehen eingeführten Rollen zumindest jeweils zwei Personen zu involvieren, um eine einseitige Darstellung der Problem- und Lösungsdomäne zu verhindern.

Gemeinsames Modellierung in Projekten. Zur gemeinsamen Modellierung in Projekten gehört es, einen Konsens zu finden. Abhängig davon, wie die gemeinsame Modellierung gestaltet ist, kommt es während der Erstellung eines Modells bzw. danach bei der Integration der Modelle zur Konsensbildung. Von Curtis et al. [CKI88] wurde dieser Vorgang beschrieben.

Abbildung 2.14 zeigt das Finden eines Konsens auf unterschiedlichen ontologischen Ebenen. Für ein Individuum ist ein selbst erstelltes Modell offensichtlich angemessen und richtig. Eine Gruppe, die das Modell zur Weiterarbeit benötigt, muss die verschiedenen Modelle integrieren, wobei im Laufe des Entwicklungsprozesses auch unterschiedliche Modelle entstehen können, die für eine Gruppe innerhalb des Teams angemessen und richtig sind. In der Phase, in der die Integration der verschiedenen Teile der Software (z. B. Komponenten) zusammengeführt bzw. deren Zusammenwirken definiert wird, muss ein gemeinsames Modell gefunden werden, das für

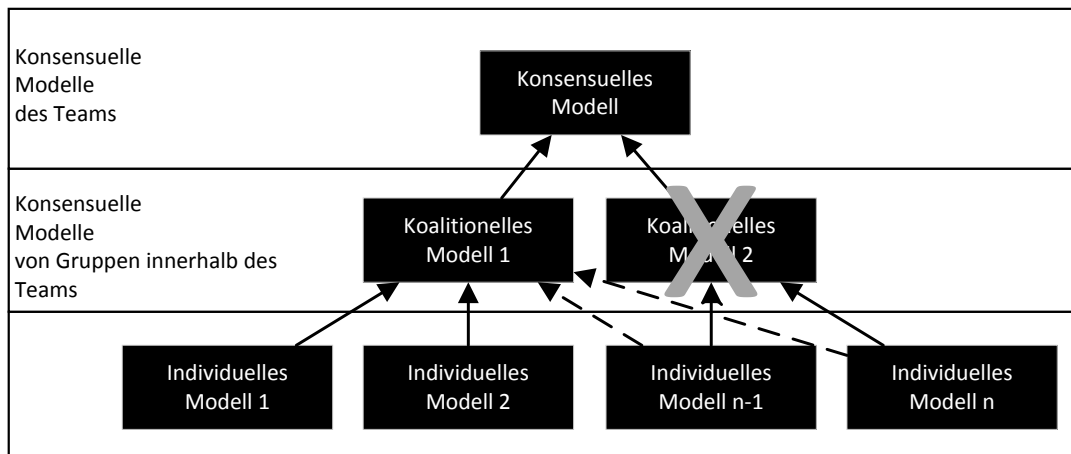


Abbildung 2.14: Design-Prozess in einer Personengruppe nach Curtis et al. [CKI88]

das gesamte Team angemessen und richtig ist. Eine Möglichkeit, dies zu erreichen, ist das Auswählen eines für alle akzeptablen Modells ohne zusätzliche Konsensfindung. Eine andere Möglichkeit ist, die als angemessen und richtig anerkannten Modelle (die alle durchaus angemessen und richtig sein können) zu einem gemeinsamen Modell zusammen zu führen. Diese Schritte sind in der Realität aber durchwegs aufwändig – z. B. aufgrund dessen, dass bereits Implementierungsarbeit erfolgt ist und daher Arbeitszeit in die Umsetzung investiert wurde. Darüber hinaus kann das gefundene konsensuelle Modell erheblich von der bereits realisierten Umsetzung abweichen und zu erheblichen Mehraufwand führen.

Eine andere Möglichkeit ist es, den initialen Aufwand der gemeinsamen Modellbildung in Kauf zu nehmen und die gemeinsame Modellierung so lange zu betreiben, bis die Arbeiten an individuelle Personen verteilbar sind. Dies ist in der Industrie aufgrund von auslastungsgetriebenen Vorstellungen der Unternehmen häufig nicht möglich. Eine Voraussetzung der gemeinsamen Modellbildung ist, dass fundiertes Domänenwissen über den Anwendungsbereich vorhanden ist, der eine konsensuelle Integration der verschiedenen Systemteile überhaupt ermöglicht.

Gemeinsame Modellierung im Modellierungsansatz dieser Arbeit. Im Entwicklungsprozess werden eine Vielzahl von Modellen erstellt. Diese Modelle werden integriert und auch wieder verworfen. Wir haben dies als nicht zielführend in dieser Arbeit verstanden und schlagen auch deswegen einen Modellierungsansatz vor, bei dem die Modelle, die in frühen Phasen des Requirements Engineerings erstellt werden, auf einem gemeinsamem Basismodell basieren. Dies soll den Integrationsaufwand reduzieren und die Möglichkeit, dass *falsche* Artefakte erstellt werden reduzieren, da alle Artefakte die in frühen Phasen im Requirements Engineering erstellt werden, konform zum Basismodell sein müssen.

2.2.5 Die Rolle von Modellen in der Softwareentwicklung

Traditionell wird Software nach einem aufeinander aufbauenden Prozess entwickelt. Programmierer schreiben Programmcode der (mehr oder weniger) konform den An-

forderungsspezifikationen und beschrieben in Modellen ist [Lid11]. Liddle [Lid11] weist darauf hin, dass es in diesem Prozess immer wieder zu Transformationen kommt, die unvollständig, umständlich und informal sind. U.a. ist daher die Idee hinter modellgetriebener Softwareentwicklung, dass die Modelle zu mehr als zur Kommunikation und der allgemeinen Verständlichkeit verwendet werden sollen und die Basis für daraus automatisch (oder semi-automatisch) generierte Software (oder Teilsystemen) sind [Lid11].

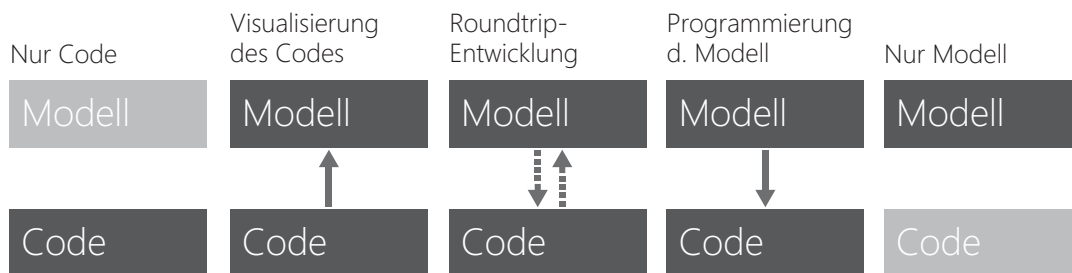


Abbildung 2.15: Spektrum der Modellierung, angelehnt an [Lid11]

Modelle kommen in unterschiedlichen Formen in der Design- und Implementierungsphase vor. Abbildung 2.15 zeigt die unterschiedliche Rolle von Modellen in den genannten Phasen in einem Softwareentwicklungsprojekt.

- ▶ **Nur Code:** In der Design- und Implementierungsphase spielen Modelle keine Rolle. Der Programmcode wird ohne Zutun von Modellen von Programmierern erstellt.
- ▶ **Visualisierung des Codes:** Modelle werden zur Visualisierung des erstellten Programmcodes verwendet um die Struktur bzw. das Verhalten abstrakter und daher für Programmierer leichter verständlich darzustellen. Hier sei angemerkt, dass auch Programmcode selber als Modell verstanden werden kann.
- ▶ **Roundtrip Entwicklung:** Modelle und Programmcode existieren nebeneinander. Modelle werden verwendet um teilweise den Programmcode zu generieren und der Programmcode wird teilweise in Modellen dargestellt. Zusätzlich dienen Modelle zur Kommunikation und Förderung der Verständlichkeit für die Programmierer. In der Praxis ist diese Modellverwendung in der Softwareentwicklung von Informationssystemen häufig anzutreffen (siehe Abschnitt 2). Teile des Programmcodes (z. B. Strukturen) werden durch Modelle generiert, zusätzlich werden z. B. Verhaltensanweisungen im Programmcode in partielle Klassen hinzugefügt. Werden Anweisungen bzw. Struktur-Informationen in den Klassen geändert wirkt sich das auch wieder auf die Modelle aus. Gutes Beispiel hierfür ist das Generieren von Datenzugriffsabstraktionsklassen aus Modellen des ADO.NET Entity Frameworks [Ler10].
- ▶ **Programmierung durch Modell:** Aus Modellen wird Programmcode generiert oder Modelle werden durch einen Interpreter interpretiert. Die Programmierer erstellen in der Implementierungsphase ausschließlich Modelle. Die Modelle bilden dabei die ausführbare Basis. Ein Beispiel ist hierfür ist AUTOFOCUS [HSS96]. AUTOFOCUS stellt ein CASE-Entwicklungswerkzeug für eingebettete Systeme nach dem Stand aktueller Wissenschaft dar. Es deckt den kom-

pletten Entwicklungsprozess vom Erheben der Anforderungen über das Systemdesign bis hin zur Code Generierung für Ausführungsplattformen ab und basiert auf einer formalen Theorie zur Spezifikation von verteilten Systemen, FOCUS [BS01].

- ▶ **Nur Modell:** Modelle werden nur für die Design-Phase verwendet. Es wird daraus weder Programmcode generiert noch wird ein solcher damit visualisiert. Modelle dienen nur dem Verstehen und Kommunizieren.

Die Roundtrip-Entwicklung, sowie die Programmierung durch Modelle stellen eine Verwendung von Modellen dar, die mit der Tätigkeit der Programmierung vergleichbar ist, aber auf einer höheren Abstraktionsebene stattfindet. Aus den Modellen wird ein ausführbarer Programmcode generiert oder durch Interpretation durch einen Interpreter werden die Modelle direkt ausgeführt. Durch die Hebung der Abstraktionsebene ist das ausführbare Modell intuitiver zu verstehen, wie es der Programmcode ist.

Einordnung des Basismodells in diese Darstellung. Das im Rahmen der Vorgehensweise in dieser Arbeit erstellte Modell, das wir Basismodell nennen, ist ein Modell, aus dem initial nichts generiert wird. Wir finden uns daher im Bereich *Nur Modell*, der Abbildung 2.15 wieder. Das Basismodell dient als Ausgangsbasis für die Detaillierung der dargestellten Inhalte in anderen Modellierungstechniken. Wir sehen das Basismodell daher als Startpunkt für einen Softwareentwicklungsprozess, der unabhängig davon ist, ob aus Modellen Programmcode generiert wird, bzw. Modelle interpretiert werden, oder ob der Programmcode manuell implementiert wird. Die Erarbeitung eines Verständnisses über die Problem- und Lösungsdomäne muss im Rahmen beider Ansätze geschehen.

2.3 Ontologien

Der Ontologie-Begriff in seiner ursprünglichen Verwendung – als die Lehre des Seienden – ist Teil der Metaphysik⁵. Ontologien werden in unterschiedlichen Disziplinen eingesetzt und innerhalb der Disziplinen wieder unterschiedlich verwendet. Dieser Abschnitt klärt den Ontologie-Begriff für diese Arbeit.

Dazu wird in Abschnitt 2.3.1 die Frage geklärt, was eine Ontologie ist und für welche Zwecke sie erstellt wird, in Abschnitt 2.3.2. Abschnitt 2.3.3 erläutert die verschiedenen Arten von Ontologien, die in der Informatik Anwendung finden. Darauf aufbauend, wird in Abschnitt 2.3.4 das ontologische Übereinkommen besprochen, wie es im Sinne dieser Arbeit verwendet wird. Ontologien werden wie andere Modelle nicht (nur) ad-hoc, sondern nach einer Vorgehensweise bzw. Methodik erstellt. Diese Erstellungsmethodiken werden in Abschnitt 2.3.5 vorgestellt.

⁵ Metaphysik behandelt grundlegende Probleme der Philosophie u.a. Fundamente, Voraussetzung, Ursachen oder (allererste) Gründe allgemeiner Strukturen, Gesetzmäßigkeiten, Prinzipien sowie Sinn bzw. Zweck allen Seines [Zal12].

2.3.1 Definition

Der Ontologie-Begriff wird in unterschiedlichen Disziplinen verwendet. Guarino et al. [GG95] identifizierten u.a. folgende Verwendungen:

- ▶ Ontologie als philosophische Disziplin
- ▶ Ontologie als ein informales, konzeptuelles Modell
- ▶ Ontologie als formale Semantik
- ▶ Ontologie als Spezifikation einer Konzeptualisierung

Neches et al. [NFF⁺91] fassen allgemein zusammen und bezeichnen eine Ontologie als die Definition von grundlegenden Begriffen und Beziehungen. Ontologien bilden einen Wortschatz eines Themengebietes, sowie die Regeln und Bedingungen für die Kombination von Begriffen und Beziehungen zur Erweiterung dieses Wortschatzes.

Bertram [Ber05] versteht unter einer Ontologie in der Informatik im Teilbereich der künstlichen Intelligenz ein formal definiertes System von Dingen und/oder Konzepten und Relationen zwischen diesen Dingen. Zusätzlich enthalten Ontologien – zumindest implizit – Regeln.

Japser et al. [JU99] definiert Ontologie in der Informatik allgemeiner: Eine Ontologie mag verschiedene Formen haben, aber notwendigerweise enthält sie immer ein Vokabular, bestehend aus Termen und deren (informale oder formale) Spezifikationen. Darüber hinaus inkludiert eine Ontologie die Definition und Spezifikation, wie Konzepte in Beziehung stehen und gibt in der Gesamtheit vor, wie die Domäne strukturiert ist und engt eine mögliche Interpretation der Terme ein [JU99].

Über diese Verwendungen von Neches et al. [NFF⁺91] hinaus gibt es spezielle Definitionen für Teilbereiche der Informatik. Eine für diese Arbeit gültige generische Definition wird im Folgenden erläutert:

Begriffsdefinition Ontologie. Die bekannteste Definition einer Ontologie in der Informatik stammt von Gruber [Gru95b, Gru93] und wurde später durch Borst [Bor97] erweitert. Eine Ontologie bezeichnet demnach eine formale, explizite Spezifikation einer von einer Gruppe geteilten Konzeptualisierung.

Wobei mit Konzeptualisierung ein abstraktes Modell eines real existierenden Phänomens (etwas beobachtbares, wahrnehmbares) gemeint ist, mit der Einschränkung, dass die relevanten Konzepte dieses Phänomens identifiziert wurden [SBF98]. Explizit bedeutet, dass die verwendeten Konzepte und die einzuhaltenden Bedingungen bei der Verwendung explizit spezifiziert sind; formal bezieht sich darauf, dass Ontologie von Computern (Maschinen) lesbar sein müssen und geteilt deutet darauf hin, dass das in Ontologien enthaltene Wissen konsensuelles Wissen ist – es ist kein privates (ein für nur einer Person gedachtes), sondern Wissen, das von einer (Personen-)Gruppe akzeptiert und geteilt wird bzw. wurde [SBF98, SS09].

Formalisierungsgrad einer Ontologie. Über diese Einteilungen und Definitionen hinweg, stellt u.a. Uschold et al. [UG96] fest, dass Ontologien auch nach ihrem Formalisierungsgrad eingeteilt werden können und definiert dazu folgende Einteilun-

gen:

- ▶ Informal: Ontologien spiegeln die natürliche Sprache wieder. Es gibt keine Vorgaben.
- ▶ Semi-informal: Strukturierte und eingeschränkte Form der Sprache, Konzepte sind nicht spezifiziert.
- ▶ Semi-formal: Strukturierte Form, Konzepte sind durch andere Ontologien oder durch Konzeptbeschreibungen spezifiziert.
- ▶ Formal: Strukturierte Form, Konzepte und deren Beziehungen sind durch eine formale Semantik spezifiziert.

Formalisierungsgrad der Ontologie verwendet in dieser Arbeit. In dieser Arbeit ist das durch den Modellierungsansatz erstellte Basismodell eine formale Ontologie. Die Konzepte und Beziehungen sind formal spezifiziert. Wir stellen darüber hinaus die Konsistenz des Modells durch Axiome sicher.

2.3.2 Gründe für die Verwendung von Ontologien

Ontologien werden nicht zum Selbstzweck erstellt. Für Uschold et al. und Noy et al. [UG96, NM01] sind Gründe zur Entwicklung einer Ontologie:

- (a) **Das Teilen eines gemeinsamen Verständnisses und Einigkeit über die Struktur von Information:** Das Teilen und das Schaffen eines gemeinsamen Verständnisses ist eines der wichtigsten Ziele der Entwicklung von Ontologien [Mus92, Gru93, Fen03]. Das Teilen bezieht sich (a) auf den maschinellen Austausch von Informationen über die Grenzen eines lokalen Computersystems hinweg, mit Hilfe von Upper-Level-Ontologien (siehe Abschnitt 2.3.3) und (b) auf das Erarbeiten und Teilen eines konsensuellen Wissens einer Gruppe von Menschen.
- (b) **Kommunikation:** Reduktion von Missverständnissen unter Personen in der Verwendung einer einheitlichen Begriffswelt. Ontologien spielen hier eine wichtige Rolle. Wenn miteinander Kommunizierende sich zu einer Ontologie bekennen, also ein ontologisches Übereinkommen schließen, können sie den Interpretationsraum über das Kommunizierte einschränken, wie es z. B. Koschmider et. al [KO05] für Geschäftsprozesse machen.
- (c) **Interoperabilität:** Unterstützung des Informationsaustausches zwischen verschiedenen Gruppen verschiedener Domänen (z. B. von Menschen, aber auch Software-Systemen). Die Ontologie enthält z. B. Meta-Informationen und unterliegt den Strukturierungsvorgaben von Meta-Ontologien, wie z. B. Dublin Core [DW03].
- (d) **Spezifikation, Wiederverwendung:** Unterstützung des Systementwicklungsprozesses durch Unterstützung von Softwaredesign-Entscheidungen sowie Implementierung durch domänenspezifisches Wissen das in Ontologien enthalten ist. Darüber hinaus können Ontologien verwendet werden, um die Spezifikation von Softwaresystemen in Form einer Ontologie gegen das Design bzw. Implementierung zu verifizieren.

- (e) **Die Wiederverwendung von Domänenwissen:** Wiederverwendung ist in der Literatur ein häufig genannter Grund zur Erstellung von Ontologien [WSBK08]. Z. B. wird im Produktlinienmanagement mit dem Ziel der Wiederverwendung von Anforderungen auf Ontologien zurückgegriffen um anwendungsübergreifendes Domänenwissen zu erfassen [LSR07]. Häufig ist unter Wiederverwendung von Domänenwissen auch strukturelles Wissen über die Begriffswelt einer bestimmten Domäne gemeint [NM01].
- (f) **Die Explizieren von Annahmen einer bestimmten Domäne:** Annahmen bzw. Sachverhalte einer Domäne lassen sich mit Ontologien explizieren. Ontologien sind einfacher zu verstehen für Nicht-Entwickler, als Programmcode und können dadurch auch als gemeinsame Repräsentation der Annahmen bzw. Sachverhalte einer Domäne gesehen werden [NM01].
- (g) **Die Trennung von Domänenwissen und operationalem Wissen:** Noy et al. [NM01] erwähnten, dass dies ein weiterer häufig anzutreffender Zweck bei der Verwendung von Ontologien ist. Z. B. trennen McGuinness et al. [MW98] ein Software-System von seiner Konfiguration und Rothenfluh et al. [RGE⁺94] verwenden Domänen-Ontologien und Methoden-Ontologien (siehe Abschnitt 2.3.3) zur Steuerung eines Aufzugs.
- (h) **Zur Analyse von Domänenwissen:** Ein wichtiges Anwendungsgebiet von Ontologien ist die Analyse von Domänenwissen. Dies inkludiert z. B. die Unterordnung einer Anwendungsbereichs-Ontologie unter eine Domänen-Ontologie [MFRW00], aber auch die Erarbeitung des relevanten (für einen bestimmten Zweck) Domänenwissens einer noch nicht erfassten Domäne [AGP04].

Gründe für die Verwendung von Ontologien in dieser Arbeit. Der in der Auflistung unter Punkt (a) angeführte Punkt, das Teilen eines gemeinsamen Verständnisses und Einigkeit über die Struktur von Information, ist ein Hauptmotiv für die Verwendung von Ontologien in dieser Arbeit. Wir haben uns für Ontologien entschlossen, da deren dargestellte Informationen einfach lesbar sind ohne den komplexen Syntax einer Modellierungssprache lernen zu müssen. Informationen können somit schnell und einfach geteilt werden. Informationen werden immer auf die gleiche Art und Weise dargestellt. Damit eignen sich die erstellten Ontologien auch sehr gut zur Kommunikation, was als Grund unter Punkt (b) angeführt wird. Wir sehen das im Rahmen der Arbeit erstellte Basismodell auch als Spezifikation der Problem- und Lösungsdomäne, die dann im Entwicklungsprozess als Basis für weitere Modelle anderer Modellierungstechniken weiter verwendet wird. Diese Modelle müssen dann konform zum Basismodell sein, dies wird unter Punkt (d) erläutert. Das Basismodell repräsentiert auch explizierte Annahmen der in die Erstellung involvierten Stakeholder, dies ist unter Punkt (f) angeführt.

2.3.3 Arten von Ontologien

Ist der Zweck für den eine Ontologie erstellt wird identifiziert, muss eine geeignete Detaillierung der Inhalte der Ontologie gewählt werden, bevor diese zu erstellen ist. Fensel [Fen03], Guarino [Gua98, Gua97] und Gómez-Pérez et al. [AGP04] identifizierten folgende Arten von Ontologien, die speziell für den Gebrauch von Ontologien als

Wissensdatenbank geeignet sind:

- ▶ **Top-Level-Ontologien oder Upper-Level-Ontologies:** Sind generische, allgemeingültige Ontologien die Raum, Zeit, Materie, Objekte, Ereignisse, Aktionen, ... beschreiben [Gua97]. Diese Ontologien sind unabhängig von einer bestimmten Domäne. Diese Ontologien können verwendet werden, um eine gemeinsame semantische Basis für unterschiedliches Wissen zu verwenden. Ein Beispiel einer solchen Ontologie ist die *IEEE SUO* (Standard Upper Ontologie) [PN02]. Eine solche Ontologie ermöglicht das Teilen von Wissen über sprachliche und räumlich Grenzen hinweg.
- ▶ **Anwendungsbereichs-Ontologien:** Enthält das Vokabular das notwendig ist für eine bestimmte Anwendung [AGP04, HSW97]. Anwendungsdomänen erweitern häufig Top-Level-Ontologien oder Domänen-Ontologien.
- ▶ **Domänen-Ontologien:** Domänen-Ontologien sind Ontologien die allgemeiner sind als Anwendungsbereichs-Ontologien. Sie können in unterschiedlichen Anwendungsbereichen einer bestimmte Domäne (wieder-)verwendet werden [MNV02].
- ▶ **Metadaten-Ontologien:** Sind strukturbeschreibende Ontologien, z. B. Dublin Core [DW03] beschreibt das Vokabular für Informationsquellen.
- ▶ **Repräsentative Ontologien:** Gehören nicht zu bestimmten Domänen. Solche Ontologien beschreiben repräsentative Konzepte ohne Aussagen darüber zu treffen, welches konkrete Wissen damit ausgedrückt werden soll [Fen03]. Diese Ontologien bieten Konzepte um die Grammatik und den Syntax von weiteren Modellierungssprachen in Form von Ontologien zu beschreiben. Ein Beispiel einer solchen Ontologie ist die Frame-Ontologie [Gru93].
- ▶ **Methoden-Ontologien:** Beschreibt das Vokabular einer Methodik bzw. eines Prozesses für das Erreichen eines bestimmten Zieles oder für eine bestimmte Tätigkeit [AGP04].
- ▶ **Tätigkeit-Ontologien:** Beschreibt das Vokabular für eine generische Tätigkeit [AGP04].

Verwendete Art der Ontologie in dieser Arbeit. Ontologien, wie sie in dieser Arbeit verwendet werden, enthalten Gegenstandstypen als auch deren konkrete Ausprägungen. Dabei entspricht die Terminologie der Gegenstandstypen einer Domänen-Ontologie (siehe dazu Kapitel 6) und die konkreten Gegenstandsmengen einer Anwendungs-Ontologie. Die Anwendung ist dabei auf folgende Domänen beschränkt: Entwicklungsprojekte für Informationssysteme in Unternehmen, die ihre Tätigkeit an Geschäftsprozessen ausgerichtet haben.

Ontologien in Art und Formalisierung im Bezug auf die Verwendung. Die Abbildung 2.16 zeigt das Ergebnis einer Literaturrecherche über die Verwendung von Ontologien, die im Rahmen dieser Arbeit durchgeführt wurde. Die Literatur findet sich im Rahmen dieses Abschnittes (2.3) an geeigneter Stelle.

Abbildung 2.16 zeigt die vom Autor der Arbeit am häufigsten in der Literatur gefundenen Detaillierungsgrade und Formalisierungsgrade. Die Abbildung ermöglicht es,

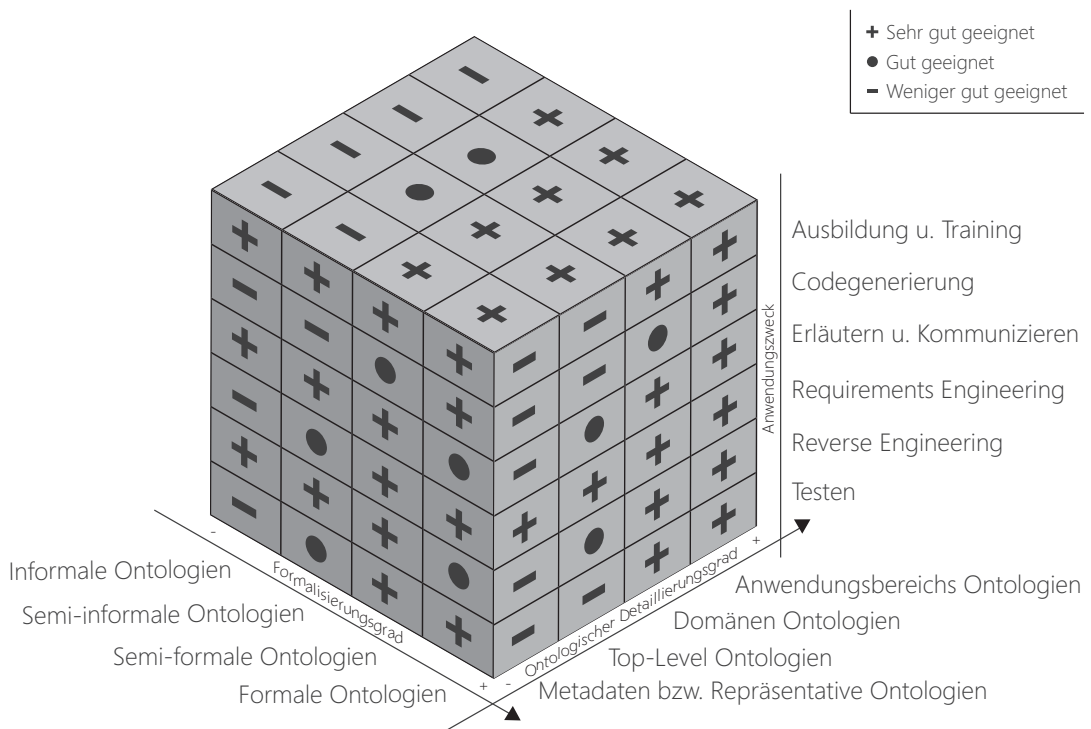


Abbildung 2.16: Formalisierungsgrad von Ontologien in Anbetracht ihrer Eignung für verschiedene Detaillierungsgrade und Anwendungszwecke

immer zwei der drei Dimensionen in Beziehung zu setzen. Die Abbildung ist daher wie folgt beispielsweise zu lesen: Für den Anwendungszweck Requirements Engineering, sind alle Arten von Ontologien sehr gut geeignet, jedoch sind für diesen Anwendungszweck nur semi-formale und formale Ontologien in ihrem Formalisierungsgrad sehr gut geeignet.

Die Bewertung wurde vom Autor der Arbeit mit Hilfe der Literaturrecherche sowie den im Rahmen der Arbeit erstellten Ontologien und den daraus gewonnene Erfahrungen durchgeführt. Die Achse Formalisierungsgrad wurde nach den Arbeiten von Uschold et al. [UG96] eingeteilt. Die Anwendungsgebiete wurden den operativen Zielen der Modellierung von Domänen nach Iscoe [IWA91] entnommen. Die dargestellten ontologischen Detaillierungsgrade wurden vom Autor nach Häufigkeit ihrer Nennung in der Literatur ausgewählt.

Die Formalisierungsgrade und die Detaillierungsgrade von Ontologien sind in Abbildung 2.16 eingezeichnet. Abhängig von der Abstraktheit der Begriffe in einer Ontologie, sowie deren Beziehungen untereinander, gibt es, wie in Abschnitt 2.3.3 im Detail beleuchtet, verschiedene Arten von Ontologien, die sich in ihrem Detaillierungsgrad unterscheiden. In Softwareentwicklungsprojekten eignen sich besonders Anwendungsbereichs-Ontologien dazu das Wissen für eine Anwendung darzustellen. Anwendungsbereichs-Ontologien sind in ihrer Verwendung auf eine bestimmte Anwendung zugeschnitten, dies bedeutet, die verwendeten Begriffe sind spezifisch genug sie einer konkreten Anwendung zuordnen zu können.

In frühen Phasen des Softwareentwicklungsprozesses eignen sich besonders semi-

formale Ontologien zur Repräsentation des relevanten Domänenwissens, z. B. wie bereits erwähnt, als strukturiertes Begriffsnetz mit Beschreibungen. Formale Ontologien können hier ebenso verwendet werden, z. B. und durch ihre zusätzlichen Regeln und Axiome Überprüfungen der Konsistenz von modellierten Sachverhalten ermöglichen. Ontologien, sofern sie auf eine Anwendung zugeschnitten sind, können daher einfach der Anwendung überführt werden, unabhängig davon, ob ihr Formalisierungsgrad semi-formal oder formal ist. Je abstrakter die taxonomische Struktur in einer bereits definierten Ontologie ist, desto schwieriger ist es, die Validität einer Ontologie zu überprüfen. Aussagen sind dann nicht mehr allgemeingültig formulierbar.

2.3.4 Ontologisches Übereinkommen

Das ontologische Übereinkommen (engl. *Ontological Commitment*) bezeichnet in dieser Arbeit:

- ▶ Das Übereinkommen der Stakeholder, die konsensuell erstellte Terminologie in der Ontologie angebracht (stimmig und konsistent) zu verwenden [GO94] und dazu notwendig
- ▶ sich zur Bedeutung der Terminologie in der Ontologie zu bekennen.

Um sich zur Bedeutung einer Terminologie einer Ontologie bekennen zu können, ist es häufig notwendig, konsensuelles Wissen zu erfassen. Hierfür ist der Ontologie-Erstellungsprozess häufig eine kooperative Tätigkeit, in die verschiedene Personen an eventuell verschiedenen Lokationen beteiligt sind. Ziel ist, dass sich alle Personen zu dieser Ontologie bekennen (engl. *commit themselves to an ontology*) [Fen03].

Nach Gruber [Gru95a] ist eines der Ziele bei der Ontologie-Erstellung, das ontologische Übereinkommen. Die Bedeutungsübereinkunft sollte für so wenig Konzepte wie möglich notwendig sein. Dies soll den Austausch von Wissen fördern [Gru95a].

Die Rolle des ontologisches Übereinkommens in dieser Arbeit. Im Rahmen der Vorgehensweise des Modellierungsansatzes ist das ontologische Übereinkommen von besonderer Bedeutung. Dazu versuchen die an der Erstellung des Basismodells beteiligten Personen eine ontologische Übereinkunft eines erhobenen Begriffes anhand der zur Verfügung gestellten Konzepte zu identifizieren und dann zu erfassen. Dies wird detailliert in Abschnitt 7.3 erläutert.

2.3.5 Entwicklungsprozess

Der Ontologie-Entwicklungsprozess ist in drei Hauptaktivitäten zerlegt. Er wurde ursprünglich von Fernández-López et al. [FLGPJ97] entwickelt und erweitert, bzw. präzisiert von Gómez-Pérez et al. [AGP04].

Gómez-Pérez et al. [AGP04] beschreiben die Hauptaktivitäten wie folgt:

- ▶ Management: Umfasst all jene Verwaltungsaktivitäten, die unter Planung, Kontrolle und Qualitätssicherung fallen.

- ▶ Entwicklung: Umfasst alle Aktivitäten, inklusive Vor- und Nacharbeiten, rund um die Entwicklung bzw. Erstellung einer Ontologie.
- ▶ Unterstützung: Umfasst alle Aktivitäten, die die Erstellung bzw. Verwaltung unterstützen.

Der ontologische Entwicklungsprozess in dieser Arbeit. Teilbereiche des Managements, hierzu gehören z. B. Maßnahmen zur Qualitäts- und Konsistenzsicherung des Basismodells werden in Abschnitt 7.3 erläutert. Die Auswahl der involvierten Rollen, daher Vorbereitungsaktivitäten, werden in Abschnitt 7.2 eingeführt. Die Entwicklung wird in Abschnitt 7.3 auf Basis der ontologischen Basis in Kapitel 6 erläutert.

2.3.6 Entwicklung

Ontologie-Entwicklung ist eine Teilaktivität des Ontologie-Entwicklungsprozesses. Die Schritte der Entwicklung von Ontologien wurden aus verschiedenen Quellen zusammengestellt und zu einem Vorgehen zusammen gefasst, [NM01, UG96, FL99, AGP04, BOS06].

Die Ontologie-Entwicklung ist eine von Subjektivität getriebene Tätigkeit, Noy et al. [NM01] stellten dazu einige Sachverhalte fest:

1. Es gibt keinen *richtigen* Weg das Wissen einer beliebigen Domäne zu modellieren, es gibt immer Alternativen. Wie gut eine Lösung ist, hängt fast immer von der Anwendung ab. Darüber hinaus spielen die eventuell angedachte Erweiterungsmöglichkeiten der Ontologie bei der Lösung eine wichtige Rolle.
2. Ontologie-Entwicklung ist notwendigerweise ein iterativer Prozess: Ontologien-Erstellung bedeutet die Phasen der Erfassung mehrfach zu durchlaufen. Ontologien werden durch das Erarbeiten eines konsensuellen Wissens häufig geändert.
3. Konzepte in einer Ontologie sollen an Objekten (physisch oder logisch) und deren Beziehungen auf den abzubildenden Anwendungsgebieten ausgerichtet sein. Häufig werden dafür Substantive (als Konzepte) und Verben (als Beziehungen) zur Beschreibung verwendet.

Erstellen, bzw. Erfassen einer Ontologie bedeutet im Allgemeinen (1) die Identifikation von relevanten Konzepten und Beziehungen unter diesen in einer bestimmten Domäne, (2) die Erfassung einer Spezifikation für die Konzepte und Beziehungen und (3) Identifikation von Begriffen die solche Konzepte und Beziehungen referenzieren sowie (4) die Akzeptanz aller beteiligten Personen über das Erfasste [UG96].

1. **Fragekatalog:** Themenbereich und Umfang festlegen (vorläufige Planung), zu berücksichtigen sind:
 - ▶ Klassen,
 - ▶ Themenbereich und
 - ▶ Verwendungszweck.

2. **Betrachtung schon bestehender Ontologien:** Es existieren für unterschiedlichste Domänen bereits eine Vielzahl von Ontologien die elektronisch zur Verfügung stehen und als Wissensbasis weiter verwendet werden können.
3. **Wichtige Begriffe/Bezeichnungen und deren Eigenschaften sammeln:** Zur Unterstützung des Zwecks einer Ontologie müssen relevante Begriffe, mit dem Ziel eine Liste von relevanten Begriffen zu erhalten, gesammelt werden (ohne Rücksicht auf Beziehungen zwischen Begriffe, Klassen oder Eigenschaften).
4. **Klassen definieren:** Klassen bzw. Kategorien⁶ definieren, in die die identifizierten relevanten Begriffe eingeordnet werden können. Dazu kann aus drei Vorgehensweisen gewählt werden:
 - ▶ Top-Down: Oberste Klassen werden definiert, danach werden auf Basis der Begriffe weitere Unterklassen gebildet, bis alle Begriffe klassifizierbar sind – die Klassifizierung erfolgt später.
 - ▶ Bottom-Up: Ausgehend von den spezifischen Begriffen werden allgemeine Klassen gebildet und die Hierarchie von den Begriffen ausgehen erstellt – auch in diesem Fall erfolgt die Klassifizierung der einzelnen Begriffe später.
 - ▶ Kombination: Ist eine Kombination aus dem Top-Down und Bottom-Up-Ansatz.

Für alle diese Vorgehensweisen muss eine geeignetes Abstraktionsniveau der Klassen durch den Zweck vorsezifiziert werden. Ziel ist es nicht, Begriffe in besonders viele Klassen zu klassifizieren, sondern sie für den dem jeweiligen Zweck dienlichen Klassen zu klassifizieren.

5. **Eigenschaften den Klassen zuordnen:** Die gefundenen Klassen bzw. Kategorien haben Eigenschaften, die sie an weitere Klassen bzw. konkreten Begriffen vererben können. In diesem Schritt werden den Klassen bzw. Kategorien diese Eigenschaften zugewiesen.
6. **Werte bzw. Ausprägungen der Eigenschaften festlegen:** Den Eigenschaften (in Form von Beziehungen) können in diesem Schritt Klassen bzw. Kategorien zugeordnet werden.
7. **Instanzen für Klassen festlegen:** Abhängig vom Zweck der Ontologie können nun die konkreten Begriffe den gebildeten Klassen bzw. Kategorien zugeordnet werden.

Nach erfolgter Erstellung muss die Ontologie evaluiert werden, um festzustellen, ob sie ihren Zweck erfüllen kann, Ansätze dazu werden in Abschnitt 2.3.8 vorgestellt.

Berücksichtigung dieser Sachverhalte im Modellierungsansatz. Diese geschilderten Sachverhalte werden im Vorgehen in Abschnitt 7.3 berücksichtigt. Das Vorgehen zur Erstellung des Basismodell ist ein iteratives Vorgehen. Die Konzepte sind nach dem Zweck (dem Anwendungsgebiet) ausgerichtet, siehe dazu Kapitel 6. Die von Uschold und Gruninger [UG96] erläuterten Schritte zur Erstellung von Ontolo-

⁶ Wir verstehen in diesem Kontext unter einer Klasse bzw. einer Kategorie einen Begriff, der eine Menge von ungefähr, in bestimmten Eigenschaften, gleichartigen Objekten repräsentiert [Dud11].

gien finden sich in der Vorgehensweise, in Abschnitt 7.3, wieder. Der angeführte Fragenkatalog findet sich im Schritt *Erhebung durch Kompetenzfragen* in Abschnitt 7.3.1 wieder.

2.3.7 Entwicklungs-Kriterien

Eine Ontologie soll effektiv die intentionale Bedeutung der definierten Terme kommunizieren [Gru95a]. Die Definitionen sollen so objektiv wie möglich sein, z. B. erweitert durch formale Axiome, alle Definitionen sollen in natürlicher Sprache dokumentiert bzw. spezifiziert werden [Gru95a].

Gruber [Gru95a] erarbeitet und präsentiert folgende Kriterien für gute Ontologie-Entwicklung (gekürzt):

- (a) **Klarheit:** Eine Ontologie soll die intentionale Bedeutung der definierten Terminologie effektiv kommunizieren. Das explizierte Wissen soll unabhängig von sozialen und technischen Kontexten sein. Alle Definitionen sollen in natürlicher Sprache definiert werden. Axiome sollen verwendet werden, wo dies möglich ist.
- (b) **Kohärent:** Eine Ontologie soll kohärent sein, die beschriebenen Konzepte sollen in ihrem Zusammenhang in natürlicher Sprache beschrieben sein und ggf. mit Beispielen. Schlussfolgerungen sollen in einer Ontologie wieder zu konsistenten Definitionen führen.
- (c) **Erweiterbarkeit:** Eine Ontologie soll so erstellt werden, dass eine Erweiterung im Vorhinein vorgesehen ist, damit im Nachhinein einfach Erweiterungen vorgenommen werden können. Es sollte für eine Reihe von vorgesehenen Aufgaben bzw. Aussagen bereits eine terminologische Basis vorhanden sein, damit Benutzer diese spezialisieren können, ohne die Konsistenz und Kohärenz der Ontologie zu gefährden.
- (d) **Minimal Codierungsaufwand:** Die Terminologie soll nicht auf einer Symbolik (bzw. Syntax) beruhen die schwer zu erlernen und zu verstehen ist. Dieser initiale Aufwand soll so minimal wie möglich sein.
- (e) **Minimales ontologisches Übereinkommen:** Eine Ontologie sollte nur ein minimales ontologisches Übereinkommen voraussetzen (siehe dazu Abschnitt 2.3.4), um das Teilen von Wissen zu ermöglichen. Eine Ontologie soll so wenig wie möglich Vorgaben darüber machen, wie der Ausschnitt der Realität zu modellieren ist. Eine Möglichkeit dies zu erreichen ist die Definition nur jener Konzepte, die notwendig sind um konsistentes Wissen abzubilden.

Die Kriterien in der Vorgehensweise in dieser Arbeit. In dieser Arbeit finden die von Gruber [Gru95a] erarbeiteten Richtlinien Anwendung. Wir haben, wie in Punkt (a) beschrieben, alle Konzepte und deren Beziehungen in natürlicher Sprache beschrieben, dies geschieht in Abschnitt 6.2. Wir haben die in Punkt (b) geforderten Eigenschaften durch die Definition von Axiomen und Erläuterungen dazu in Abschnitt 6.3 berücksichtigt. Wir haben, auf Basis der in Artefakte explizierten Inhalte versucht, eine minimale Terminologie zu erarbeiten, deren Konzepte disjunkt sind, wir berücksichtigen daher auch Punkt (d). Die in der Terminologie erfassten Begriffe

sind, wie geschildert, disjunkt und die Begrifflichkeiten so gewählt, dass keine Verwechslungsgefahr ihrer Bedeutung beim Finden eines ontologischen Übereinkommens auftreten sollte, wir berücksichtigen daher auch Punkt (e).

2.3.8 Evaluierung

Neben den Design-Richtlinien zur Ontologie-Erstellung, gibt es auch Ansätze zur Evaluierung von erstellten Ontologien bzw. Ontologien die sich in der Erstellung befinden.

Gómez-Pérez et al. [GPJP95, AGP04] unterscheiden in diesem Zusammenhang folgende Begrifflichkeiten:

- ▶ **Evaluation:** Ist die technische Beurteilung der Inhalte einer Ontologie während allen Phasen des Lebenszykluses einer Ontologie. Die Evaluation inkludiert die Verifikation und Validation der Ontologie. Einzelne Bestandteile der Evaluation sind u.a. alle Individuen, Definitionen, Axiome, Definitionen aus anderen Ontologien usw. [GPFV96].
- ▶ **Verifikation:** Darunter wird die Korrektheit einer Ontologie verstanden in Anbetracht der Anforderungen an eine Ontologie z. B. durch die Definition von Zielen bzw. Kompetenzfragen.
- ▶ **Validation:** Darunter wird die Validität in Anbetracht der Abbildung der realen Welt verstanden. Fragen darin sind u.a. bildet die Ontologie die ausgewählten Ausschnitte der Realität korrekt ab?
- ▶ **Assessment:** Ist die Beurteilung der Inhalte einer Ontologie aus Sicht der Verwender. Für unterschiedliche Verwender muss dieses Assessment auch unterschiedliche Anforderungen mit in die Beurteilung mit einbeziehen.

Brank et al. [BGM05] haben versucht Ontologie-Evaluierungstechniken zu klassifizieren und haben folgende gefunden:

- ▶ **Vergleich:** Vergleich der Ontologie mit einer Standard-Ontologie (z. B. in der Domäne von Interesse anerkannte Ontologie, auch engl. *Gold Standard* genannt) [MS02].
- ▶ **Verwendung:** Durch die Anwendung der Ontologie wird das Ergebnis evaluiert [PM04].
- ▶ **Abdeckung:** Evaluierung durch die Quelle des Wissens, z. B. Menge von Dokumenten, und der Abdeckung des Wissens darin [BADW04].
- ▶ **Angebrachtheit:** Die Ontologie wird durch die Ersteller bzw. Verwender nach bestimmten Kriterien bewertet [LTGP04].

In der Evaluation einer Ontologie kann diese primär von zwei Personengruppen wahrgenommen werden, dem Ontologie-Ersteller(n) und den Domänenexperten, die nachfolgend erläutert werden.

Evaluierung durch Ontologie-Ersteller. Evaluierungskriterien einer Ontologie können sein [GPFV96, GP99, BOS06]:

- ▶ **Konsistenz:** Konsistenz einer Ontologie hat mehrere Seiten, u.a. kann von einer strukturellen Konsistenz gesprochen werden, u.a. wenn die taxonomische Struktur der Konzepte keine Zyklen hat. Des weiteren ist eine inhaltliche Konsistenz dann gegeben, wenn keine semantischen Widersprüche auftreten durch die Definition von Konzepten und deren Begriffen und auch keine axiomatischen Widersprüche gefunden werden. Darüber hinaus ist eine Ontologie dann konsistent via Schlussfolgerung, wenn sich nur Wissen ableiten lässt, dass wieder zu einer konsistenten Ontologie führt.
- ▶ **Vollständigkeit:** Unvollständigkeit ist ein fundamentales Problem in Ontologien. Die Vollständigkeit einer Ontologie kann nicht bewiesen werden, eine Ontologie kann dennoch als vollständig angesehen werden, wenn alles Wissen, welches vorgesehen ist in der Ontologie explizit definiert oder sich via Schlussfolgerung eruieren lässt [AGP04]. Zusätzlich kann eine Ontologie als unvollständig gesehen werden, wenn z. B. nicht alle Begriffe Konzepten zugeordnet sind oder wenn Aufteilungsfehler der Begriffe (z. B. durch falsche Klassifizierung) vorliegen. Die Vollständigkeit ist ebenfalls bedroht von Konzepten mit überschneidender Bedeutung von Begriffen, die ebenfalls diesem Konzept zugeordnet sind. Eine mögliche Lösung, ist die Erstellung von disjunkten Konzepten.
- ▶ **Redundanz:** Redundanz ist u.a. dann gegeben, wenn synonyme Begrifflichkeiten gleichen Konzepten zugeordnet sind. Dies ist eine grammatische Redundanz. Darüber hinaus kann eine taxonomische Redundanz vorhanden sein, wenn das vorher genannte auf Ebene der Konzepte geschieht.

Eine weitere Form der Redundanz ist Überschneidung der Bedeutung von Konzepten in der Ontologie. Dies kann z. B. verhindert werden, indem das überschneidende Konzepte so lange verfeinert wird, bis disjunkte Konzepte möglich sind. Die überschneidenden Konzepte müssen aus der Ontologie entfernt werden.

Evaluierung durch Domänenexperten. Evaluierung durch Domänenexperten bedeutet die inhaltliche Bewertung der Ontologie nach verschiedenen Metriken. U.a. wird in [BOS06] gezeigt, wie durch eine automatische und manuell erstellte Ontologie nach verschiedenen Metriken inhaltlich bewertet werden kann, die u.a. in [LTGP04] erarbeitet wurden.

Wichtiger Aspekt hierbei ist die inhaltliche Bewertung von Konzepten, Beziehungen sowie der taxonomischen Struktur der Konzepte und Beziehungen, sowie den Axiomen. Hierfür wird eine Skala von eins bis fünf (sehr hoch) eingesetzt um einzelne Kriterien der Aspekte zu bewerten bzw. quantitativ zu repräsentieren.

2.4 Verwandte Arbeiten

In dieser Arbeit stellen wir eine Methode vor, um das Überbrücken der Kommunikationslücke in frühen Phasen des Requirements Engineerings zu unterstützen.

Dies wird in der Wissenschaft sehr unterschiedlich betrachtet. Für uns bedeutet

das Überwinden der Kommunikationslücke, das Erarbeiten eines gemeinsamen Verständnisses. Dies gilt auf Seiten der Softwareentwickler wie auch auf Seiten der Domänenexperten bzw. anderen relevanten Stakeholdern.

Wir betrachten in diesem Abschnitt den Stand der Wissenschaft und ebenso Ansätze, die dafür entwickelt wurden, um in frühen Phasen des Requirements Engineerings eingesetzt zu werden. Dies trifft vor allem auf Analyse-Ansätze zu. Wir betrachten in diesem Kapitel überblicksmäßig Analysemuster (in Abschnitt 2.4.1), die objektorientierte Analyse (in Abschnitt 2.4.2) und die Geschäftsprozessanalyse (2.4.3), die wir miteinbezogen haben, da diese in frühen Phasen verwendet werden kann, um die Problemdomäne besser zu verstehen. Abschließend betrachten wir einen Ansatz der allgemeinen Unternehmensabbildung (2.4.4), der verwendet werden kann, um verschiedene Eigenschaften einer Unternehmung in einer gesamtheitlichen Sicht abzubilden.

Alle diese Analyse-Techniken stellen wir in Bezug zu den in Kapitel 5 vorgestellten Problemen durch das Einschätzen der Adressierung der Probleme.

Wir fassen die Ergebnisse und weitere verwandte Arbeiten in Abschnitt 2.4.5 zusammen.

Wir betrachten in diesem Abschnitt nicht die Erhebungs- und Repräsentationstechniken des allgemeinen Wissensmanagements. Diese sind losgelöst von einer konkreten Verwendung und adressieren ohne Methodik keine der adressierten Probleme. In Abschnitt 2.1.1 in Abbildung 2.5 stellen wir wichtige Vertreter der Wissensrepräsentationstechniken vor. Die Methodik der Erhebung wird für eine ausgewählte Wissensrepräsentationstechnik in Abschnitt 2.3.5 behandelt.

Analyse-Techniken. Der Analyse-Vorgang ist wichtige Voraussetzung für die Entwicklung eines Softwaresystems. Die Domäne, in der das Softwaresystem operiert, muss von Softwareentwicklern verstanden werden, damit die Informationen der Domäne geeignet in der Software abgebildet und von dieser unterstützt werden können. In der Analyse müssen sich die Analysierenden so genau mit einer Domäne beschäftigen, dass auch Nuancen entdeckt werden, die selbst Domänenexperten noch nicht in Betracht gezogen haben [CY94].

In diesem Abschnitt führen wir bekannte Analyse-Techniken ein. Die vorgestellten Analyse-Techniken werden in frühen Phasen des Requirements Engineerings eingesetzt, um das für ein Softwareprojekt relevantes Wissen zu erheben.

Wir unterscheiden eine datenorientierte und prozessorientierte Sicht auf eine Domäne und stellen jeweils wichtige Vertreter vor, die auf die Analyse für eine datenorientierte bzw. prozessorientierte Sicht abzielen.

2.4.1 Analysemuster

In Geschäftsbereichen kommen häufig ähnliche Strukturen vor. U.a. wird von Fowler [Fow99] mit Analysemustern versucht, dabei zu helfen diese ähnlichen Strukturen zu erkennen und zu modellieren. Die Analysemuster helfen den Modellierern bzw. Analysten dabei wiederkehrende, zu analysierende Ausschnitte aus einem Geschäftsbereich auf eine bewährte Art und Weise zu modellieren. Ziel ist es u.a. die

Analyse eines Geschäftsbereiches zu beschleunigen und ähnliche Sachverhalte unterschiedlicher Geschäftsbereiche auf ähnliche Art und Weise zu modellieren.

Von Zhao und Verschuereen [ZV07] werden diese wiederverwendbaren Muster Problemmuster genannt. Das Ziel ist in beiden Fällen das gleiche; das Verstehen und Explizieren von Domänenwissen durch Analyse und Strukturierung desselbigen. Die durch diese Analyse-Methoden erstellten Modelle entsprechen Domänenmodellen.

Beispiel eines Analyseusters. Das Analysemuster Umrechnungsfaktor (engl. Originaltitel *Conversion Ratio*) hilft dabei, die in der Anwendungsdomäne durch das Softwaresystem abgebildeten Mengen mit Mengeneinheiten und Umrechnungsfaktoren zu versehen. Dazu werden Mengeneinheiten mit Umrechnungsfaktoren assoziiert. Abbildung 2.17 zeigt das Muster.

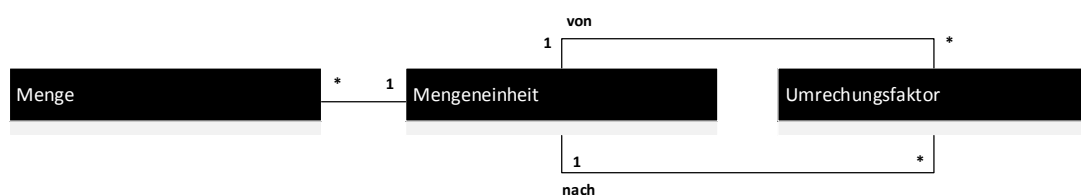


Abbildung 2.17: Das Analysemuster Umrechnungsfaktor nach Fowler [Fow99]

Für Fowler [Fow99] ist dieses Muster von großer Bedeutung. In der Softwareentwicklung werden Datentypen häufig ohne Angabe einer konkreten Mengeneinheit im Programmcode verwendet. Softwareentwickler die diese Datentypen weiter verwenden können fälschliche Annahmen darüber treffen, in welcher Mengeneinheit ein konkreter Wert eines Datentyps vorliegt. Dies kann zu Problemen im Betrieb des Softwaresystems führen. Dieses Muster expliziert eine Menge mit konkreter Mengeneinheit und ggf. mit Umrechnungsfaktoren in andere Einheiten (z. B. die Längeneinheiten Kilometer in Meilen und zurück).

Adressierung der Probleme anhand der Problemstellung. Die Einstufung der Art der Adressierung der Probleme erfolgt nach den erhobenen Problemen in Kapitel 3, 4 und 5.

Nachfolgende Tabelle erläutert die Adressierung:

Problem bzw. Erläuterung	A.
Problem 1: Inhaltliche Missverständnisse: Das Problem von inhaltlichen Missverständnissen soll adressiert werden, indem für häufig auftretende ähnliche Probleme Repräsentationen vorgegeben werden.	●
Problem 2: Untersch. Abstraktionsebenen von Artefakten: Dieses Problem wird teilweise durch die Muster adressiert. Z. B. wird festgelegt, dass die Beziehung zwischen Organisationseinheiten modelliert wird. Auf die unterschiedliche Abstraktionsebene zwischen den entstehenden Artefakten wird nicht eingegangen.	●
Problem 3: Keine Konsistenzsicherung von Artefakten: Die Muster sollen bereits konsistente Vorlagen für die repräsentierten Inhalte sein. Darüber hinaus ist keine Konsistenzsicherung vorgesehen.	●

Problem bzw. Erläuterung	A.
Problem 4: Unterschiedliche Modellbildung: Die Muster adressieren die Modellbildung insofern, dass sie diese vorweg nehmen indem sie Schablonen anbieten. Auf zusätzliche Eigenschaften einer Domäne kann dadurch nicht eingegangen werden.	●
Problem 5: Unausgeglichene Repr. von Wissen: Dieses Problem wird nicht direkt adressiert. Die Muster, die es für verschiedene Probleme gibt, können aber auch als Schablonen dafür gesehen werden, welches Wissen in einem Problembereich auftreten kann und behandelt werden sollte.	●
Problem 6: Keine Verknüpfung zu Artefakten: Problem wird nicht adressiert.	○
Problem 7: Kein stakeholderweites Verständnis: Die Muster adressieren das Problem insofern, dass sie die Kommunikation unter den Personen erleichtern, die über die Muster und den verwendeten Syntax Wissen haben.	●

Legende: A.: Adressierung, ●: Problem wird adressiert, ○: Problem wird teilweise adressiert, ○: Problem wird nicht adressiert.

2.4.2 Objektorientierte Analyse

In der objektorientierten Analyse (OOA) ist es das Ziel, die Objekte (bzw. Konzepte) in einem (Problem-)Bereich (Domäne) zu finden und zu beschreiben [Lar05]. In der Analyse wird versucht das Problem zu analysieren, nicht die Lösung. Die objektorientierte Analyse ist eine Untersuchung der Domänen-Objekte [Lar05].

Häufig wird in der Literatur auch von objektorientierter Analyse und Design (OOAD) gesprochen. Der Design- bzw. Entwurfsteil von OOAD betont die konzeptuelle Lösung, die die Anforderungen erfüllt [Lar05]. Die konzeptuelle Lösung ist dann Ausgangsbasis für die Implementierung der Lösung.

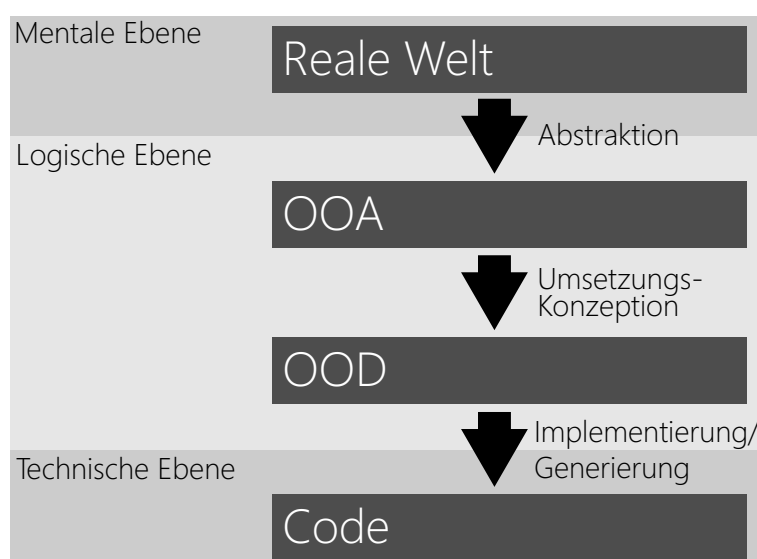


Abbildung 2.18: Abstraktionsebenen involviert in der OOAD

Abbildung 2.18 zeigt die in der OOAD involvierten Abstraktionsebenen. Im initialen Schritt wird durch einen Abstraktionsprozess der mentalen Ebene auf die logische Ebene übergegangen. Dazu werden verschiedene Modelle der OOA eingesetzt um relevante Objekte der Realität und deren Beziehungen untereinander zu identifizieren und zu explizieren. Dies involviert eine datenorientierte und dynamische Sicht auf die relevante Domäne. Von den OOA-Modellen wird auf die Entwurfs- bzw. Design-Modelle übergegangen. Diese Modelle dienen dem Übergang zur physischen Ebene durch Implementierung des Programmcodes und/oder durch Generierung des Programmcodes durch die Modelle.

Die OOAD folgt dabei typischerweise folgenden Schritten [JCA92, Lar05]:

- ▶ **Analyse – Anwendungsfälle definieren:** In den Anwendungsfällen wird die Interaktion des Softwaresystems aus der Perspektive von Akteuren niedergeschrieben.
- ▶ **Analyse – Domänenmodell definieren:** In dieser Phase wird eine Beschreibung der Domäne mit Hilfe von Objekten erstellt [Lar05]. Dazu werden Konzepte, Attribute und Assoziationen die für die Aufgabenstellung als relevant erachtet werden, identifiziert [Lar05].
- ▶ **Design – Interaktionsdiagramme definieren:** In diesem Arbeitsschritt werden Kollaborationsbeziehungen expliziert (z. B. mit Hilfe von Sequenzdiagrammen) [Lar05]. Konkret wird der Austausch von Nachrichten zwischen Objekten veranschaulicht [Lar05]. Dieser Schritt entspricht der Erstellung einer dynamischen Sicht auf das System.
- ▶ **Design – Designklassendiagramme definieren:** In diesem Arbeitsschritt wird die statische Sicht auf das System entworfen. Augenmerk wird auf Attribute, Methoden und Klassen gelegt [Lar05].

Hier sei erwähnt, dass diese Vorgehensweise der verhaltensorientierten Vorgehensweise entspricht. Im Umfeld der OOAD wird in der Literatur auch die datenorientierte Vorgehensweise [RBP⁺91] propagiert. Der Unterschied liegt in der Reihenfolge der durchgeführten Schritte. In der verhaltensorientierten Vorgehensweise werden zuerst die Abläufe festgehalten und in der datenorientierten Vorgehensweise die Klassen.

Adressierung der Probleme anhand der Problemstellung. Die Einstufung der Art der Adressierung der Probleme erfolgt nach den erhobenen Problemen in Kapitel 3, 4 und 5.

Nachfolgende Tabelle erläutert die Adressierung:

Problem bzw. Erläuterung	A.
Problem 1: Inhaltliche Missverständnisse: Die Modelle der OOAD werden erstellt, um diese inhaltlichen Missverständnisse zu klären bzw. Anforderungen und andere Artefakte zu spezifizieren. Sie adressieren dieses Problem nicht direkt. Inhaltliche Missverständnisse werden auf einer syntaktischen Ebene adressiert.	●
Problem 2: Untersch. Abstraktionsebenen von Artefakten: Problem wird nicht adressiert.	○

Problem bzw. Erläuterung	A.
Problem 3: Keine Konsistenzsicherung von Artefakten: Die Konsistenzsicherung ist auf eine syntaktische Ebene begrenzt.	●
Problem 4: Unterschiedliche Modellbildung: Problem wird nicht adressiert.	○
Problem 5: Unausgeglichene Repr. von Wissen: Problem wird nicht adressiert.	○
Problem 6: Keine Verknüpfung zu Artefakten: Eine mögliche Verknüpfung von Artefakt-Typen auf andere Artefakt-Typen ist nicht spezifiziert. Die unterschiedlichen Artefakt-Typen bieten aber die Möglichkeit dies zu tun.	●
Problem 7: Kein stakeholderweites Verständnis: Das Problem wird nicht direkt adressiert, indirekt durch die erstellten Modelle.	●

Legende: A.: Adressierung, ●: Problem wird adressiert, ○: Problem wird teilweise adressiert, ○: Problem wird nicht adressiert.

2.4.3 Geschäftsprozessanalyse

Mielke [Mie01] argumentiert, dass Softwaresysteme die in Geschäftsbereichen agieren, häufig im Rahmen der Verbesserung von Geschäftsprozessen eingesetzt werden. Dies führt uns zur Erkenntnis, dass es Teil der Domänenanalyse sein sollte auch Geschäftsprozessanalyse zu betreiben und den gewünschten Soll-Zustand der abzubildenden Geschäftsprozesse zu explizieren.

Den Soll-Zustand der Geschäftsprozesse zu modellieren bedeutet sich über die Erwartungen bzw. Ziele klar zu werden, u.a. sind das nach Becker et al. [BKR08]: Einsparung von Kosten, Straffung von Arbeitsabläufen, Reduktion von Planungszeiten und Verkürzung von Bearbeitungszeiten.

Ausgehend vom Ziel und Zweck der Modellierung muss zuerst festgelegt werden, in welchem Umfang die Modellierung erfolgen soll [BKR08]. In der Softwareentwicklung müssen Geschäftsprozesse vielfältig unterstützt werden, angefangen von der Prozesslandschaft (der Übersicht der Geschäftsprozesse) bis hin zu feingranularen Arbeitsschritten eines Geschäftsprozesses.

In diesem Abschnitt konzentrieren wir uns auf Erhebungsmethoden, die im Geschäftsprozessmanagement eingesetzt werden, um relevanten Geschäftsprozesse zu identifizieren. Im Rahmen der Analyse müssen die erhobenen Geschäftsprozesse anhand von verschiedenen Kriterien bewertet werden und daraus werden die Soll-Geschäftsprozesse abgeleitet. Die Kriterien sind abhängig von den Zielen und Zwecken, die mit den Geschäftsprozessen erreicht werden sollen. Wir können daher unter Geschäftsprozessanalyse den Vorgang der Erhebung als Mittel der Analyse bzw. dem Verstehen der Geschäftsprozesse verstehen oder/und aber auch das Bewerten anhand von Kriterien von bereits explizierten Geschäftsprozessen. Für Beyer [Bey03] inkludiert die Prozessanalyse das Erkennen der Ist-Situation ebenso wie auch die Analyse um Schwachstellen zu erkennen und die Prozesse auf Verbesserungspotentiale hin zu überprüfen.

Die Analyse der Geschäftsprozesse in der Softwareentwicklung folgt der Aussage,

dass die Analyse Mittel zum Erkennen der Geschäftsprozesse und die Software Mittel zur Unterstützung zum Erreichen der Soll-Situation ist.

Methoden der Erhebung. Beyer [Bey03] hat einige auf eigene Erfahrung beruhende Erhebungsmethoden mit ihren Vor- und Nachteilen zusammengestellt, die wir in ihren Ausführungen gekürzt und erläutert haben:

- ▶ **Interview mit Manager:** Vorteile sind u.a.: Kompetenz, Kenntnis der Mitarbeiter, geringer Zeitbedarf (nur eine Person pro Abteilung). Nachteile sind u.a.: Das Soll-Denken von Managern, die Realitätsfremde und die Tatsache, dass Manager meist keine detaillierten Kenntnisse der einzelnen Aktivitäten haben.
- ▶ **Interview mit Mitarbeitern:** Vorteile sind: Realitätsnahe Angaben und die Möglichkeit zur exakten Beschreibung der Tätigkeit durch die Angaben der Mitarbeiter. Nachteile sind: Eventuelle mangelnde Kompetenz der Mitarbeiter und der Zeitaufwand (durch die Befragung mehrerer Mitarbeiter pro Abteilung).
- ▶ **Fragebogen:** Vorteile: Viele Daten können in kurzer Zeit erhoben werden und, die Ergebnisse liegen schriftlich vorliegen. Nachteile: Mangelhafter Rücklauf und ungenaue Daten sind möglich.
- ▶ **Workshop:** Vorteile eines Workshops sind: Dass Lösungsansätze gleich mit angedacht werden können und dass schneller als in Einzelinterviews Erhebungen stattfinden können. Nachteile: Dass auch der Workshop zeitaufwändig ist und mehrere Mitarbeiter gleichzeitig von der Arbeit abgehalten werden.
- ▶ **Eigene Beobachtung:** Vorteile: Eigene Beobachtungen sind unabhängig und subjektiv und realitätsnah. Mitarbeiter werden kaum bei der Arbeit gestört. Nachteile: Zeitaufwändig, und die mangelnde Kenntnis der betriebsinternen Abläufe durch einen betriebsfernen Beobachter.
- ▶ **Analyse bereits vorhandenen Datenmaterials:** Vorteile: Geringer Zeitbedarf und teilweise sind die Ausgangsmaterialien bereits aufbereitet. Nachteile: Art, Vollständigkeit und Glaubwürdigkeit der Daten ist eventuell unbekannt und die Daten können veraltet sein.

Erweiterte ereignisgesteuerte Prozessketten. Die allgemeine Geschäftsprozessanalyse kann nur mit einer gewählten Repräsentationsform bewertet werden. Ergebnis der Analyse ist ein Modell einer Geschäftsprozessmodellierungstechnik. Nach Morelli [Mor10] ist in Deutschland die EEPK-Methode⁷ etabliert, die wir als Repräsentationstechnik der Analyse in der Bewertung verwenden.

Die ereignisgesteuerten Prozessketten (EPK) bilden den Kern der ARIS-Methode. ARIS steht für Architektur integrierter Informationssysteme. Die EPKs bilden die zentrale Beschreibungssprache der ARIS-Methode [Sch02, Leh07]. In der Anwendung ist ARIS streng mit dem Werkzeug der IDS Scheer AG verbunden, dadurch gibt

⁷ Erweiterte ereignisgesteuerte Prozessketten (EEPK) ist eine Sprache zur Modellierung von Geschäftsprozessen und Arbeitsabläufen und ist wichtiges Element des ARIS-Konzeptes (Architektur Integrierter Informationssysteme) [SN00].

es keine Organisation, die die Beschreibungssprache verbindlich vorgibt und/oder weiterentwickelt – es fehlt daher an einem Standard [All09, Mor10].

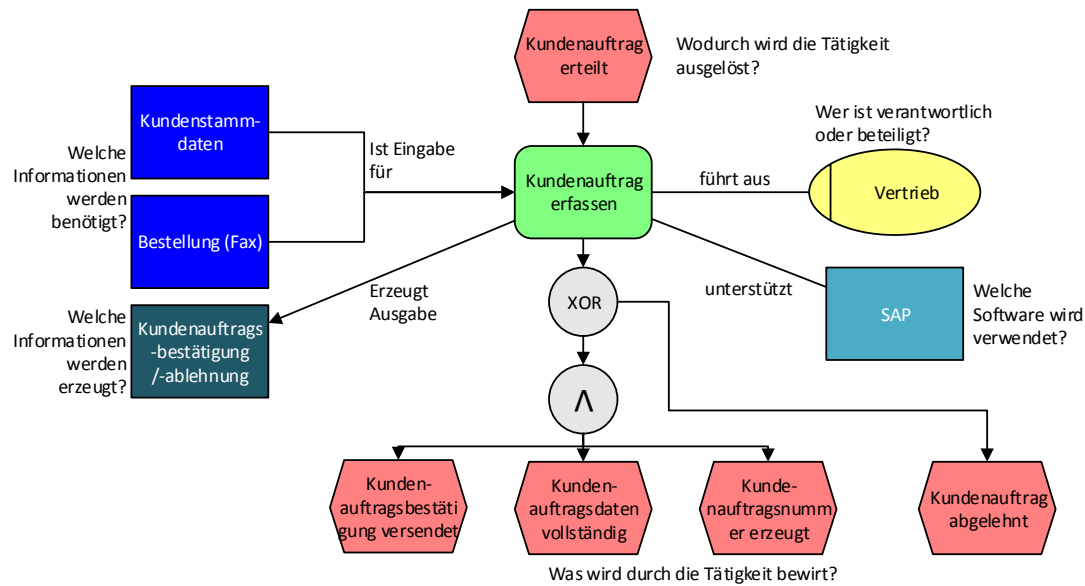


Abbildung 2.19: Beispielhaftes EEPK-Modell, angelehnt an das einführende Beispiel von [Mor10] mit erklärenden Fragestellungen

Die Abbildung 2.19 zeigt ein Beispiel eines Geschäftsprozesses, modelliert mit der Beschreibungssprache EEPK. Die einzelnen Modellelemente sind mit Fragen annotiert. Durch die Beantwortung dieser Fragen können die Entitäten erhoben werden, die im Modell repräsentiert werden.

Adressierung der Probleme anhand der Problemstellung. Die Einstufung der Art der Adressierung der Probleme erfolgt nach den erhobenen Problemen in Kapitel 3, 4 und 5.

Nachfolgende Tabelle erläutert die Adressierung:

Problem bzw. Erläuterung	A.
<p>Problem 1: Inhaltliche Missverständnisse: Die Analyse von Geschäftsprozessen dient, wie anfänglich im Abschnitt erwähnt, verschiedenen Zielen. U.a. auch der Klärung von inhaltlichen Missverständnissen. Dies bezieht sich aber nur auf die Aktivitäten in Geschäftsprozessen bzw. auf die Geschäftsprozesse selber. Die involvierten Geschäftsobjekte werden dabei nur informal eingebunden.</p>	●
<p>Problem 2: Untersch. Abstraktionsebenen von Artefakten: Problem wird nicht adressiert.</p>	○

Problem bzw. Erläuterung	A.
Problem 3: Keine Konsistenzsicherung von Artefakten: Die Artefakte der ARIS-Methode folgen dem Syntax der Beschreibungssprache <i>EEPK</i> . Die Konsistenz der Artefakte wird auf syntaktischer Ebene sichergestellt. In Verbindung mit Prozesslandschaften auch übergreifend. Daraus lassen sich auch Aussagen zur inhaltlichen Konsistenz treffen. Dies gilt aber nur für die Geschäftsprozesse und Aktivitäten und nicht für die Ressourcen und involvierte Geschäftsobjekte.	●
Problem 4: Unterschiedliche Modellbildung: Problem wird nicht adressiert.	○
Problem 5: Unausgeglichene Repr. von Wissen: Dieses Problem wird nicht direkt adressiert. Wird von einer Prozesslandschaft ausgegangen, ist es das Ziel die Einzelprozesse zu spezifizieren. Die Übergänge in der Prozesslandschaft wären dann auch expliziert. Werden alle Prozesse der Prozesslandschaft spezifiziert, dann ist das Wissen auch zumindest initial ausgeglichen modelliert.	●
Problem 6: Keine Verknüpfung zu Artefakten: Verknüpfungen sind nur zu den Modellierungskonzepten der ARIS-Methode vorhanden.	●
Problem 7: Kein stakeholderweites Verständnis: Dies wird nicht adressiert. Die Prozesslandschaft ist aber ein Einstiegspunkt zur Erarbeitung eines gemeinsamen Verständnisses.	●

Legende: A.: Adressierung, ●: Problem wird adressiert, ○: Problem wird teilweise adressiert, ○: Problem wird nicht adressiert.

2.4.4 Unternehmensabbildung

Wir betrachten in diesem Abschnitt einen Ansatz zur gesamtheitlichen Unternehmensabbildung (Englisch *Enterprise Modelling*) der eine starke ontologische Abdeckung bietet. Eine starke ontologische Abdeckung ist dann gegeben, wenn ausgewählte Ausschnitte der Wirklichkeit in einer Modellierungssprache eindeutig dargestellt werden können, unter der Annahme, dass die Modellierungssprache die Wirklichkeit repräsentieren will [FL03]. Ein Ansatz, auf den dies zutrifft ist der Ansatz des Semantischen Objektmodells (SOM) von Ferstl und Sinz [FS90, FS91, FS95, FS06].

SOM ist ein Ansatz zur Modellierung von betrieblichen Systemen. Der Ansatz erstreckt sich auf das gesamte betriebliche Objektsystem⁸. Um die Komplexität eines betrieblichen Systems beherrschbar zu machen, wird das Objektsystem in Teilsystemmodelle unterteilt. Dabei beschreiben die Teilsystemmodelle jeweils das Objektsystem unter einem bestimmten Blickwinkel. Die Teilsystemmodelle und ihre Beziehungen bilden die Unternehmensarchitektur [FS06].

SOM unterscheidet drei Modellebenen eines betrieblichen Systems [Sin97]: (a) den Unternehmensplan als Modell der Außensicht, (b) Geschäftsprozessmodelle als Mo-

⁸ Ein relevanter Ausschnitt der Realität (die Diskurswelt) mit dazugehörigem nicht zur Diskurswelt gehörendem Ausschnitt der Umwelt wird Objektsystem genannt. Wobei die Diskurswelt offen und zielgerichtet ist. Dabei besteht das Objektsystem aus interagierenden betrieblichen Objekten, welche die Aufgaben des Objektsystems durchführen [FS95].

dell der Innensicht und (c) Modelle der Ressourcen zur Durchführung von Geschäftsprozessen. Wobei die Ebenen (b) und (c) durchgängig objektorientiert modelliert werden.

Unterschiede in der Modellierung zum vorgestellten Modellierungsansatz in dieser Arbeit. Im in dieser Arbeit vorgestellten Modellierungsansatz modellieren wir für ein gemeinsames Verständnis unter den Stakeholdern wichtige Begriffe. Dabei werden die Begriffe ontologischen Konzepten zugeordnet um ihre Bedeutung festzulegen. Zusätzlich werden stereotypisierte Beziehungen zwischen den Begriffen erfasst. In diesem Abschnitt heben wir wichtige Unterschiede zwischen SOM und den in dieser Arbeit vorgestellten Modellierungsansatz vor.

Geschäftsprozesse involvieren Geschäftsobjekte. In der Analyse (siehe Kapitel 4) haben wir festgestellt, dass die Art der Involviertheit von Geschäftsobjekten wesentlicher Betrachtungsgegenstand in Artefakten ist, die in frühen Phasen erstellt wurden. In SOM wird diese Involviertheit durch die Benennung von Beziehungen festgelegt. Im Modellierungsansatz, den wir in dieser Arbeit vorgestellt haben, haben wir vier verschiedene Arten der Involviertheit von Geschäftsobjekten in Geschäftsprozessen im Rahmen der Analyse gefunden. Im SOM Vorgangs-Ereignis-Schema wird die Veränderung eines Geschäftsobjektes nicht betrachtet. Hier stellt der in dieser Arbeit vorgestellte Modellierungsansatz darüber hinaus ontologische Konzepte zur begrifflichen Repräsentation von Zuständen von Geschäftsobjekten zur Verfügung.

Im strukturorientiertem konzeptuellen Objektschema im SOM-Ansatz werden die konzeptuellen Objekttypen eines Anwendungssystems und ihre Beziehungen erfasst. Eine Stereotypisierung der Objekttypen wird nicht vorgenommen. Die repräsentierten Objekttypen werden daher nur durch ihre Bezeichner und den Beziehungen in ihrer Bedeutung eingeschränkt. Im in dieser Arbeit vorgestellten Modellierungsansatz wenden wir neben der umfangreicheren Anzahl von Beziehungen zusätzlich ontologische Konzepte zur Bedeutungseinschränkung von Begriffen an. In [MA99] wird festgestellt, dass die Klassifizierung, die wir durch die ontologischen Konzepte erreichen, wesentliches Instrument des Menschen zur Bedeutungsfindung ist. Dieser Aspekt wird in SOM nicht gesondert betrachtet.

Die Modellbildung spielt im Ansatz in dieser Arbeit eine wichtige Rolle. Das Basismodell wird im Rahmen eines Konstruktionsprozesses erstellt (siehe Kapitel 7). Der Konstruktionsprozess sieht u.a. einen Dialog zwischen den relevanten Stakeholdern vor. Im Rahmen dieses Dialoges wird das Basismodell konstruiert. Diese Vorgehensweise führt u.a. dazu, dass das erfasste Wissen im Basismodell ausgeglichen repräsentiert ist. Das heißt, Ausschnitte der gewählten Diskurswelt bzw. deren Umwelt sind nicht über- oder unterrepräsentiert. SOM sieht hier keine Vorgehensweise in der Modellbildung einzelner SOM-Modelle vor.

Adressierung der Probleme anhand der Problemstellung. Die Einstufung der Art der Adressierung der Probleme erfolgt nach den erhobenen Problemen in Kapitel 3, 4 und 5.

Nachfolgende Tabelle erläutert die Adressierung:

Problem bzw. Erläuterung	A.
Problem 1: Inhaltliche Missverständnisse: Reduktion von inhaltlichen Missverständnissen durch gesamtheitliche Darstellung (Unternehmensabbildung).	●
Problem 2: Untersch. Abstraktionsebenen von Artefakten: Adressierung durch die SOM-eigenen Modelle und deren Verknüpfung untereinander.	◐
Problem 3: Keine Konsistenzsicherung von Artefakten: Adressierung durch die SOM-eigenen Modelle.	●
Problem 4: Unterschiedliche Modellbildung: Problem wird nicht adressiert.	○
Problem 5: Unausgeglichene Repr. von Wissen: Problem wird nicht adressiert.	○
Problem 6: Keine Verknüpfung zu Artefakten: Verknüpfung zwischen Modellen nur für SOM-eigene Modelle.	◐
Problem 7: Kein stakeholderweites Verständnis: Das Problem wird nicht direkt adressiert. Die SOM-Modelle können durch die gesamtheitliche Darstellung des betrieblichen Objektsystems als Repräsentation eines stakeholderweiten Verständnisses gesehen werden.	◐

Legende: A.: Adressierung, ●: Problem wird adressiert, ◐: Problem wird teilweise adressiert, ○: Problem wird nicht adressiert.

2.4.5 Zusammenfassung, Stand der Wissenschaft

In diesem Abschnitt haben wir Analyse-Techniken behandelt, die in frühen Phasen des Requirements Engineerings eingesetzt werden können um ein gemeinsames Verständnis zu erarbeiten und so die Kommunikationslücke zu überbrücken.

Keiner der vorgestellten Ansätze behandelt die erhobenen Probleme direkt. Dies ist deswegen besonders irritierend, da hinreichend anerkannt ist, dass das Problemverständnis ein kritischer Erfolgsfaktor für Softwareentwicklungsprojekte ist [Off02].

In der Literatur werden viele Arbeiten im Vorfeld, z. B. Ziele und Stakeholder identifizieren, Ziele klassifizieren, Systemumfang festlegen [Par10] erwähnt. Bei den Erhebungstechniken werden von Partsch [Par10] ebenso Techniken aufgezählt, u.a. Kreativitätstechniken, Beobachtungstechniken, Befragungstechniken, Evolutionstechniken usw. Alle erhobenen Informationen müssen auch expliziert werden.

Trotz der in [Par10] angeführten Studie, dass 31% der Fehler in Anforderungen darauf zurückzuführen sind, dass Anforderungen fehlen oder inadäquat sind, 49% darauf, dass die Anforderungen inkorrekt und 18% darauf, dass sie nicht konsistent und/oder unklar sind, decken die uns bekannten Techniken zur Modellierung nur isolierte Ausschnitte bzw. Sichten auf eine Problemdomäne ab und die Integration wird von vielen der Ansätze überhaupt nicht betrachtet.

Hier sei darauf hingewiesen, dass es viele weitere Analyse-Ansätze gibt (z. B. Domänenanalyse im Produktlinienmanagement [LSR07]), die aber typischerweise nicht in den frühen Phasen des Requirements Engineerings eingesetzt werden und daher nicht in diesem Abschnitt behandelt wurden.

Dies gilt auch für Ansätze der Unternehmensabbildung. In der Unternehmensabbildung wird versucht, verschiedene Sichten auf ein Unternehmen zu offerieren und teilweise zu integrieren. Dies involviert typischerweise die Aufbau- wie Ablauforganisation, Geschäftsobjekte und Anwendungslandschaften und noch weitere Sichten. Diese Ansätze sind nicht dafür gemacht in frühen Phasen des Requirements Engineerings das Wissen einer Problem- und Lösungsdomäne zu erheben. Sehr wohl aber können sie dieses Wissen zum Teil repräsentieren. Dabei wird in der Unternehmensabbildung aber typischerweise eine höhere Abstraktionsebene angestrebt. Die Entwicklung eines Softwaresystems steht nicht im Mittelpunkt. Im Mittelpunkt dieser Betrachtungen stehen Softwaresysteme, die die Teilbereiche der Geschäftstätigkeiten unterstützten. Dies aber, wie erwähnt, auf einer hohen Abstraktionsebene.

2.5 Zusammenfassung

Dieses Kapitel führt in die wichtigsten Grundlagen zum Verstehen des Modellierungsansatzes, vorgestellt in dieser Arbeit, ein. Dazu wurden in Abschnitt 2.1.1 der Wissensbegriff erläutert, die Formen in den Wissen auftreten kann expliziert und die Gründe für implizites Wissen erarbeitet. In Abschnitt 2.1.2 wurde der Domänen-Begriff erläutert sowie dessen Rolle in der Softwareentwicklung umrissen. Diese beiden Ausführungen wurden in Abschnitt 2.1 zusammengeführt und in Abschnitt 2.1.4 wird die generische Rolle von Wissen in der Softwareentwicklung dargelegt.

Ein wichtiger Aspekt in der Modellierung, die Modellbildung, wird in Abschnitt 2.2 im Detail behandelt. Der Abschnitt inkludiert wichtige Aspekte der Modellbildung u.a. den Modellbegriff (siehe Abschnitt 2.2.1), den Vorgang der Modellierung (siehe Abschnitt 2.2.2) sowie der Probleme in der gemeinsamen Modellierung (siehe Abschnitt 2.2.4).

Modellierung erfolgt mit Methoden und deren Repräsentationsformen, in Abschnitt 2.3 wurde mit der Ontologie eine grundlegendes Modellierungskonzept eingeführt. Dazu wurde via Literaturrecherche dargelegt, was eine Ontologie ist (siehe Abschnitt 2.3.1) sowie Gründe für die Erstellung dargelegt (siehe Abschnitt 2.3.2), deren Arten unterschieden (siehe Abschnitt 2.3.3), der Entwicklungsprozess beleuchtet (siehe Abschnitt 2.3.5), verschiedene Design- und Qualitätskriterien und Evaluierungsmöglichkeiten expliziert (siehe Abschnitt 2.3.7 und 2.3.8).

In Abschnitt 2.4 werden Techniken zur Analyse, die in frühen Phasen des Requirements Engineerings eingesetzt werden können dargestellt und auf Basis ihrer Adressierung der Problemstellung in Kapitel 5 kurz beleuchtet. Abschließend wird in Abschnitt 2.4.5 zusammenfassend ein Blick auf weitere verwandte Arbeiten geworfen, die in anderen Phasen des Softwareentwicklungsprozesses bzw. unabhängig von diesem, eingesetzt werden können.

Analyse der Überbrückung der Kommunikationslücke in der Praxis

Dieses Kapitel beantwortet Forschungsfrage 1:

RQ1: Wie wird die Kommunikationslücke in Softwareprojekten in frühen Phasen des Requirements Engineerings in der Praxis überbrückt?

Zur gleichen Zeit sind wir daran interessiert herauszufinden, welche Probleme bezüglich RQ1 in der Praxis auftreten können. Die Erhebung der Probleme dient der Beantwortung der Forschungsfrage 3:

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2¹ bei den Überbrückungsansätzen in der Praxis auf?

Zur Beantwortung dieser Fragen haben wir eine Studie durchgeführt, die in diesem Kapitel erläutert wird.

Beginnend klären wir in Abschnitt 3.1 den Kontext der durchgeführten Studie. Ausgewählten Probanden wurden eine Menge von offenen Fragen gestellt. Die Fragen und der Aufbau der Studie wird in Abschnitt 3.3 erläutert. Die Antworten auf die Fragen wurden zu inhaltlichen Aussagen zusammengefasst und in Abschnitt 3.4 aufgelistet. Abschließend werden die Ergebnisse in Abschnitt 3.5 analysiert. Die Validität der Studie wird in Abschnitt 3.6 erläutert.

3.1 Kontext

Wir strukturieren den Kontext der Studie nach der Einteilung nach Wohlin et al. [WRH⁺12].

Aktive Befragung. Dazu wurden acht Personen in Tiefeninterviews befragt. Tiefeninterviews zeichnen sich besonders durch Erkenntnistiefe und der Qualität der

¹ RQ2 und der Anteil von RQ2 an RQ3 wird in Kapitel 4 beantwortet.

Ergebnisse aus [KE10]. Zur Validität der Studie sei auf Abschnitt 3.6 verwiesen.

Professioneller Bereich und reale Probleme. Wir waren an der Vorgehensweise der Überbrückung der Kommunikationslücke in solchen Unternehmen interessiert, welche basierend auf einem Auftrag Informationssysteme entwickeln. Die Unternehmen entwickeln daher Individualsoftware. Typischerweise wird hierbei die Kommunikationslücke in diesen Unternehmen bei jedem Projekt neu überbrückt.

Die Erarbeitung des initialen Problemverständnisses obliegt in unserem Fall bereits den Auftragnehmern.

Spezialisierung der Befragten. Befragte Softwareunternehmen erstellen Informationssysteme u.a. für diese Branchen: Groß- und Einzelhandel, allgemeinen Dienstleistungssektor, Industrieanlagen-Management, Sozialwirtschaft, ...

Die Teamgröße der Entwicklungsteams, aus denen die befragten Teilnehmer stammen, betrug von 4 bis 15 Entwickler.

3.2 Forschungsziel

Wir legen das Forschungsziel gemäß der Vorlage von Wohlin et al. [WRH⁺12] fest:

- ▶ **Analysiert wird:** Überbrückung der Kommunikationslücke in frühen Phasen des Requirements Engineerings
- ▶ **Für den Zweck:** Verständnisgewinn
- ▶ **Im Bezug auf:** Stand der Praxis
- ▶ **Aus Sicht von:** Softwareentwicklern und Anforderungsingenieuren
- ▶ **Im Kontext:** Requirements Engineering für die Entwicklung von prozessorientierten Informationssystemen

3.3 Aufbau der Studie

Im Vorfeld treffen wir eine Annahme, die sich auf das Design der Studie auswirkt. Wir gehen davon aus, dass in irgendwelcher Form Artefakte erstellt wurden, um Erkenntnisse aus Erhebungen in frühen Phasen des Requirements Engineerings explizit zu erfassen.

Wiederholung der Definition von Seite 2:

Definition Artefakt:

Unter Artefakt verstehen wir ein Arbeitsergebnis eines Vorganges im Rahmen des Softwareentwicklungsprojektes. Ein Artefakt hat Eigenschaften (Struktur, Verhalten, usw.) und kann mit Hilfe von standardisierten (semi-)formalen Modellierungskonzepten beschrieben sein [MFPKB10]. Wir meinen daher als Artefakt jede Form von expliziertem Wissen (z. B. in Form von Modellen, Dokumenten, handschriftliche Skizzen, usw.) als Ergebnis eines Erfassungsvorganges. Der Erfassungsvorgang kann z. B. aufgrund eines Erhebungsvorganges angestoßen werden.

Wir haben uns aufgrund der erwarteten unterschiedlichen Herangehensweisen an das Thema für die Methode der Tiefeninterviews entschieden. Diese geben keine Antworten vor und beeinflussen daher auch nicht den Beantwortungsvorgang. Die Interviews wurden durch Leitfragen gesteuert. Dabei galt zu jeder Zeit: (a) jede Frage konnte offen und ohne Zeitlimitation beantwortet werden, (b) jede befragte Person konnte jederzeit Informationen hinzufügen, die auch von der Frage abwichen, (c) die befragten Personen konnten Fragen zur Klärung von Missverständnissen stellen (z. B. wenn die Frage nicht klar war) und (d) der Interviewer konnte während der Beantwortung der Leitfragen zusätzliche Fragen zur Klärung von Missverständnissen bzw. zum tieferen Verständnisgewinn stellen.

Hier sei nochmals auf den Kontext der Studie hingewiesen, dass sich alle Fragen auf Artefakte in frühen Phasen des Requirements Engineerings beziehen.

Wir haben die Fragen in vier Blöcke eingeteilt:

A Erfassungsprozess bzw. Modellbildung:

- A1 Zu welchem Zweck werden Artefakte in frühen Phasen erstellt?
- A2 Welches Wissen wird in den Artefakten erfasst?
- A3 Wie wird entschieden, welches Wissen in den Artefakten erfasst wird?
- A4 Welche Methoden werden eingesetzt um das Wissen zu erheben?
- A5 Wird nicht erfasstes bzw. fehlendes Wissen explizit erfragt?
- A6 Welche (Modellierungs-)Techniken werden zur Erfassung verwendet?
- A7 Wie häufig müssen die Artefakte verändert werden?
- A8 Wie wirken sich diese Änderungen auf andere Artefakte aus?
- A9 Wie wird die Konsistenz der Artefakte gesichert?
- A10 Wie wird die Abstraktionsebene der Artefakte bestimmt?
- A11 Welchen Umfang haben die Artefakte?
- A12 Wie häufig werden überschneidende Sachverhalte durch verschiedene Mitarbeiter erfasst?
- A13 Werden diese Sachverhalte auch gleich erfasst (werden ähnliche Artefakte erstellt)?
- A14 Wie wirkt sich eventuell unterschiedlich erfasstes Wissen aus (z. B. auf Kommunikation, Verständnis, Integration, usw.)?

B Stakeholder:

- B1 Welche Personen oder Personengruppen sind in den Erstellungsprozess von Artefakten allgemein involviert und wie werden diese identifiziert?
- B2 Wie viele und welche Personen sind bei der Erstellung von einzelnen Artefakte beteiligt?
- B3 Welche Personen oder Personengruppen verwenden diese Artefakte?
- B4 Gibt es definierte Verantwortlichkeiten für die Artefakte?
- B5 Gibt es Verständnisschwierigkeiten bezüglich den Artefakten zwischen den Stakeholdern?

C Management von Artefakten:

- C1 In welchen Phasen des Softwareentwicklungsprozesses werden die Artefakte verwendet?
- C2 Welche Probleme treten bei der Verwendung der Artefakte in späteren Phasen auf?
- C3 Mit welchen Werkzeugen werden die Artefakte erstellt?
- C4 Gibt es Vorgaben zur Verwendung mit den Artefakten?
- C5 Wie werden die Artefakte integriert?
- C6 Wie werden die Artefakte in die Anforderungsspezifikation bzw. dem Lastenheft integriert?
- C7 Welche Probleme treten bei den Integration auf?

D Wiederverwendung:

- D1 Werden die erstellten oder andere Artefakte (z. B. des Auftraggebers) in frühen Phasen des RE wiederverwendet?
- D2 Wie wird Wiederverwendung betrieben?
- D3 Gibt es Vorgaben zur Wiederverwendung?
- D4 Treten Probleme bei der Wiederverwendung auf?

Die Transkription wurde durch den Interviewer während der Interviews stichwortartig durchgeführt. Pausen, Tonhöhenbewegungen, usw. wurden dabei nicht mit transkribiert. Die stichwortartigen Antworten wurden dann nach den Interviews innerhalb der Leitfragen gruppiert. Passte eine Antwort nicht eindeutig zu einer dieser Gruppen (Antwortgruppe), wurde eine neue Antwortgruppe innerhalb der Leitfrage hinzugefügt. Die Antwortgruppen wurden aus den stichwortartigen Antworten gebildet. Anmerkungen, die keiner Antwortgruppe zuzuordnen waren, werden in den Ergebnissen separat angeführt.

3.4 Ergebnisse

Wir sprechen in diesem Abschnitt häufig von Erheber und Modellierer.

Definition Erheber:

Der Erheber ist die Rolle, die, z.B. im Rahmen eines Interviews, das Wissen erhebt, und daher auch die mentale Auswahl trifft, welches Wissen explizit erfasst wird.

Definition Modellierer:

Der Modellierer ist die Rolle, die ein Modell bzw. ein Artefakt erstellt.

Wir unterscheiden diese zwei Rollen, da der Erheber häufig auch Skizzen erstellt, die dann von einem Modellierer in ein Artefakt überführt werden. Die Rollen müssen daher nicht identisch sein, in der Praxis findet häufig eine Personalunion statt.

Wichtigstes zusammengefasstes Ergebnis der Studie ist, dass Artefakte in frühen Phasen des Requirements Engineerings ohne Vorgaben erstellt werden. Sie dienen überwiegend dem Erarbeiten eines gemeinsamen Problem- und Lösungsverständnisses und werden in der Anforderungsspezifikation integriert.

In frühen Phasen des Requirements Engineerings werden typischerweise E/R-Modelle, Klassen-, Aktivitäts- und Sequenzdiagramme und Skizzen seitens der Anforderungsingenieure erstellt. Auf Seiten der Domänenexperten werden allgemeine Modelle über Aufbau- und Ablauforganisation erfasst bzw. wiederverwendet.

Dieser Abschnitt beschreibt zusammenfassend wichtige Ausschnitte aus den Themenblöcken. In den Überschriften ist das jeweilige Kürzel des Frageblockes (z. B. A) und den Paragraphen des Kürzels der Frage (z. B. A1) in Klammern angegeben.

Die transkribierten Interviews wurden in Antwortgruppen je Leitfrage zusammengefasst. In den Ergebnissen nennen wir diese Antwortgruppen Aussagen und führen die Anzahl an Personen an, deren Antwort sich einer bestimmten Aussage zuordnen ließ.

Es wurden alle Aussagen weggelassen, die nur von Einzelpersonen getätigt wurden, sofern es keine Ergänzungen sind.

3.4.1 Erfassungsprozess bzw. Modellbildung (A)

Dieser Abschnitt fasst stichwortartig die wichtigsten Ergebnisse des Blockes Erfassungsprozess bzw. Modellbildung zusammen.

Zweck (A1):

Aussage	Anzahl
Erarbeitung einer gemeinsamen Vorstellung, des Problembereiches	8
Sicherstellung, dass vom gleichen Sachverhalt ² gesprochen wird	8
Scoping ³	2

Erfasstes Wissen (A2):

² Unter Sachverhalt verstehen wir einen abstrakten, in Aussagen vergegenwärtigten Gegenstand [Bro02]. Sachverhalte werden aus Aussagen durch Abstraktion gewonnen [Bro02].

³ Scoping leitet sich aus dem Englischen (*scope*) ab und bedeutet soviel wie die Festlegung eines Geltungsbereiches bzw. das Abgrenzen eines Bereiches.

Aussage	Anzahl
Wissen über den Problembereich (Geschäftsprozesse, Organisationsstrukturen, usw.)	8

Hier sei angemerkt, dass durchwegs von 5 Befragten auch die Vorstellung über die Lösung explizit angegeben wurde. Auf Nachfrage wurde hier angegeben, dass die Vorstellung über die Lösung von den Anforderungen häufig schwer getrennt werden kann. Da auf Basis der Anforderungen lösungsspezifische Artefakte abgeleitet werden und diese häufig bereits im Rahmen der Analyse im Requirements Engineering entstehen und die Phasen auch nicht einfach trennbar sind.

Entscheidung, welches Wissen (A3):

Aussage	Anzahl
Wird von den Auftraggebern bzw. den Domänenexperten getroffen	6
Wird auch vom Auftragnehmer getroffen (u.a. bei konkreten Nachfragen, bzw. weil angenommen wird, dass der Auftragnehmer weiß, welches Wissen er erfassen muss)	8
Es existieren keine expliziten Vorgaben	8

Methoden zur Erhebung (A4):

Aussage	Anzahl
Interviews	8
Dokumentenanalyse (falls Dokumente vorhanden)	4
Kreativitätstechniken in Gruppen (Brain-Storming, Mind-Mapping)	3
Szenarien/Use-Cases	7
Prototyping	2
Workshops	4

Erfragung von fehlendem Wissen (A5):

Aussage	Anzahl
Fehlendes Wissen kann nur in Ausnahmen erfragt werden, da das Fehlen vom Erheber nicht festgestellt werden kann	7
Fehlendes Wissen wird erst in späteren Phasen (nach dem Requirements Engineering) erfragt	6

Hier wurde angemerkt, dass fehlendes Wissen trotz mehrfacher Iterationen aus Erhebung, Modellierung/Analyse, Spezifikation und Validierung häufig nicht gefunden wird.

Techniken zur Erfassung (A6):

Tabelle 3.2 zeigt die verwendeten Techniken zur Erfassung des erhobenen Wissens, Tabelle 3.1 enthält die Legende sowie die Anzahl der Verwender der Techniken.

Die Tabelle 3.1 und 3.2 geben die Ergebnisse der verwendeten Techniken wieder. Teilweise können sich diese überschneiden. Z.B. ist es möglich, Use Cases in Word

Modellierungstechnik	Anzahl
Use Cases (UC)	7
Strukturdiagramme (Str.)	7
Sequenzdiagramme (Seq.)	4
Mind-Maps (MM)	3
Prozessdiagramme (Proz.)	5
Aktivitätsdiagramme (Aktiv.)	4
Organisationsdiagramme (Org.)	2
Skizzen (Skiz.)	4
Logische Aussagen bzw. Zustände (Log.)	2
Word	1
Excel	2

Tabelle 3.1: Anzahl und Legende der verwendete Techniken zur Erhebung der Tabelle 3.2

und/oder Excel zu erfassen.

Nr.	UC	Str.	Seq.	MM	Proz.	Aktiv.	Org.	Skiz.	Log.	Word	Excel
1	x	x	x			x		x	x		
2	x	x	x	x	x			x	x	x	x
3	x	x			x						
4		x		x	x	x	x	x			x
5	x	x	x			x					
6	x	x	x		x						
7	x			x	x		x	x			
8	x	x				x					
Anzahl	7	7	4	3	5	4	2	4	2	1	2

Tabelle 3.2: Ergebnisse verwendete Modellierungstechniken zur Erhebung eines initialen Problem- und Lösungsverständnisses

Häufigkeit von Änderungen (A7):

Aussage	Anzahl
Änderungen sind häufig notwendig	8

Auswirkung von Änderungen (A8):

Aussage	Anzahl
Auswirkungen von Änderungen auf andere Artefakte werden bei Integration in die Anforderungsspezifikation bzw. Lastenheft behandelt	7
Auswirkungen werden ggf. sofort analysiert	3
Änderungen führen zu inkonsistenten Artefakten	5
Änderungen führen zu Neuerstellung von Artefakten	3

Konsistenzsicherung (A9):

Aussage	Anzahl
Konsistenz der Artefakte untereinander wird nicht explizit gesichert	8
Konsistenzsicherung erst bei Integration in die Anforderungsspezifikation bzw. bei Erstellung des Lastenhefts	8

Es sei angemerkt, dass sehr wohl von allen Teilnehmern ausgesagt wurde, dass die Konsistenz der Einzelartefakte durch den Erheber überprüft wird. Es gibt dazu aber keine expliziten Vorgaben und auch keine Möglichkeit, dies automatisiert zu machen.

Bestimmung von Abstraktionsebenen (A10):

Aussage	Anzahl
Es gibt keine konkrete Vorgehensweise	8

Umfang von Artefakten (A11):

Unter Berücksichtigung des Mini- und Maximums der getätigten Aussagen, haben Modelle, die in frühen Phasen erstellt wurden, nach Angaben 5 bis 30 Entitäten.

Bei der durchgeführten Analyse von Inhalten von Artefakten (siehe dazu Kapitel 4) hat ein durchschnittliches Artefakt ca. 12 Entitäten.

Bei Use-Cases/Szenarien wurde von 3 Personen angegeben, dass diese eine Länge von ein bis zwei Seiten haben.

Überschneidung von Inhalten in Artefakten (A12):

Aussage	Anzahl
Überschneidung kommen in Artefakten häufig vor	2
Überschneidungen kommen kaum vor	6
Mögliche Überschneidungen sind nicht bekannt	6

Erfassung gleicher Sachverhalte (A13):

Aussage	Anzahl
Gleiche Sachverhalte werden von Mitarbeitern unterschiedlich modelliert	7

Ein Befragter merkte an, dass sich die Modelle mit der Zeit innerhalb eines Teams angleichen (z. B. wenn in einem Team ein Geschäftsprozess immer als eigene Entität dargestellt wird, wird dies über lange Sicht auch von Mitarbeitern gemacht, die das bisher nicht getan haben).

Auswirkung unterschiedlich erfasstes Wissen (A14):

Aussage	Anzahl
Kommunikationsprobleme	7
Missverständnisse	7
Probleme beim Scoping	2
Integrationspunkte sind nicht einfach identifizierbar	3
Integration von Artefakten wird häufig nur textuell beschrieben	5

3.4.2 Stakeholder (B)

In der Erstellung involvierte Personen (B1):

Aussage	Anzahl
Bei der Erstellung werden vom Auftraggeber geeignete Ansprechpartner identifiziert	5
Direkt vom Auftragnehmer benannt (z. B. wegen fehlendem Wissen über Teilbereiche)	4
Auftraggeber: Domänenexperten	8
Auftraggeber: Manager	8
Auftragnehmer: Projektleiter	6
Auftragnehmer: Anforderungs-Ingenieur/Domänen-Analyst	8
Auftragnehmer: Software-Architekt	3

Wie viele und welche Personen sind beteiligt bei einzelnen Artefakten (B2):

Aussage	Anzahl
1 bis 2 Stakeholder des Auftraggebers	8
> 2 Stakeholder des Auftraggebers	3
1 Person des Auftragnehmers	8
2 bis 3 Personen des Auftragnehmers	6
Nachbearbeitung wird von einer Person vorgenommen	5

In den Aussagen wurden mehrere Schilderungen von Befragten zusammengestellt. Die gleichen Personen, die angaben, dass ein bis zwei Stakeholder des Auftraggebers anwesend waren, gaben auch an, dass es auch vorkommt, dass mehr als zwei Personen beteiligt sind. Dies liegt daran, dass bei der Erhebung mehrere Interviews durchgeführt wurden, und die Anzahl der Teilnehmer variierte.

Verwendung der Artefakte (B3):

Aussage	Anzahl
Allen Mitarbeitern eines Projektes	8

Definierte Verantwortlichkeiten (B4):

Aussage	Anzahl
Es gibt keine explizit definierten Verantwortlichkeiten	8
In Teams wird der Ersteller eines Artefaktes implizit als Verantwortlicher gesehen	5

Verständnisschwierigkeiten zwischen Stakeholdern (B5):

Aussage	Anzahl
Artefakte nicht <i>lesbar</i> für Stakeholder	5
Artefakte widersprechen sich	6

Angemerkt wurde, dass sich Artefakte häufig deswegen widersprechen, weil zwei unterschiedliche Mitarbeiter des Auftragnehmers auf Basis von Interviews Artefakte erstellen, wobei die Befragten unterschiedliche Personen waren.

Zusätzlich wurde angemerkt, dass in der Design- und Implementierungsphase typischerweise die bis dahin nicht aufgetauchten Missverständnisse auftreten, die aufgrund von Verständnisschwierigkeiten entstanden sind. Angestrebt wird in diesem Zusammenhang eine enge Zusammenarbeit mit dem Auftraggeber.

3.4.3 Management von Artefakten (C)

Verwendung in Phasen der Softwareentwicklung (C1):

Aussage	Anzahl
Die Artefakte werden in allen Phasen des Softwareentwicklungsprozesses verwendet	8

Probleme bei Verwendung in späteren Phasen (C2):

Aussage	Anzahl
Artefakte sind mehrdeutig oder widersprechen sich (Details müssen nachgefragt werden)	7
Beschriebene Inhalte nicht klar	5
Artefakte sind losgelöst voneinander	7
Artefakte beschreiben nicht relevante Ausschnitte	3

Werkzeuge (C3):

Genannt wurden, ohne Gewichtung, da die einzelnen Tools für eine Vielzahl von Modellierungstechniken verwendet werden können: Papier und Bleistift, Visual Studio, Microsoft Visio, Visual Paradigm, Microsoft Word und Excel, Mind-Manager, usw.

Vorgaben zur Verwendung (C4):

Aussage	Anzahl
Keine Vorgaben des Unternehmens zur Verwendung	8

Integration der Artefakte (C5):

Aussage	Anzahl
Die Artefakte werden nur in Ausnahmefällen integriert	7
Artefakte sind aufgrund der unterschiedlichen Sichten nicht integrierbar	6

Integration in die Anforderungsspezifikation bzw. Lastenheft (C6):

Aussage	Anzahl
Keine Vorgaben zur Integration	8

Angemerkt wurde, dass dies immer der Person/den Personen obliegt, die die Anforderungsspezifikation bzw. das Lastenheft bearbeitet/bearbeiten.

Probleme bei Integration (C7):

Aussage	Anzahl
Integration nicht (einfach) möglich, da der Umfang der Artefakte zu groß	5
Kein Systemvorstellung anhand des Problembereiches vorhanden	6
Keine Integration aufgrund der Anzahl von Artefakten	3
Keine Integration aufgrund häufiger Änderungen von Artefakten	5
Relevanz von Artefakten ist nicht bekannt	4

3.4.4 Wiederverwendung (D)**Wiederverwendung allgemein (D1):**

Aussage	Anzahl
Wiederverwendung von Dokumenten des Auftraggebers allgemein	6
Wiederverwendung von Modellen des Auftraggebers allgemein	5
Wiederverwendung von aufbauorganisatorischen Artefakten	6
Wiederverwendung von ablauforganisatorischen Artefakten	4

Wiederverwendung operativ (D2):

Aussage	Anzahl
In der Prozessanalyse werden relevante Aktivitäten und die involvierten Stakeholder und andere Ressourcen identifiziert	2

Vorgaben zur Wiederverwendung (D3):

Aussage	Anzahl
Keine Vorgaben zur Wiederverwendung	8
Hängt vom Auftraggeber ab	4

Probleme bei Wiederverwendung (D4):

Aussage	Anzahl
Relevanz von vorhandenen Dokumenten des Auftraggebers kann nicht festgestellt werden	3
Sicht der Artefakte (z. B. Prozessbeschreibungen) sind nicht in Zusammenhang mit den zu entwickelnden System zu bringen	5

3.5 Analyse und Interpretation

Dieser Abschnitt beantwortet die Forschungsfrage 1:

RQ1: Wie wird die Kommunikationslücke in Softwareprojekten in frühen Phasen des Requirements Engineerings in der Praxis überbrückt?

Und liefert einen Beitrag zur Beantwortung der Forschungsfrage 3:

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

3.5.1 Forschungsfrage 1: Vorgehensweise in der Praxis

In diesem Abschnitt fassen wir wichtige Ausschnitte aus den Ergebnissen zusammen.

Artefakte werden in frühen Phasen vor allem zur Erarbeitung einer gemeinsamen Vorstellung des Problembereiches und zur Sicherstellung, dass vom gleichen Sachverhalt gesprochen wird, erstellt (A1). Primär wird Wissen über den Problembereich erfasst (A2). Dies involviert Geschäftsprozesse, Organisationsstrukturen, usw. Die Auswahl welches Wissen erfasst wird, wird von Auftraggebern bzw. den Domänenexperten getroffen. Noch häufiger jedoch wird diese Entscheidung auch vom Auftragnehmer getroffen. Generell existieren dazu aber keine Vorgaben (A3).

Wissen wird überwiegend durch Interviews und Szenarien bzw. Use-Cases erhoben (A4). Fehlendes Wissen kann dabei nur in Ausnahmen erfragt werden, da das Fehlen von notwendigem Wissen nicht festgestellt werden kann. Häufig wird fehlendes Wissen erst in späteren Phasen (nach dem Requirements Engineering) gefunden (A5). Artefakte, die in frühen Phasen erstellt werden, müssen häufig geändert werden (A7). Die Auswirkung von Änderungen wird großteils erst bei der Integration in die Anforderungsspezifikation bzw. dem Lastenheft behandelt. Zusätzlich führen Änderungen zu inkonsistenten Artefakten (A8). Die Konsistenz der Artefakte wird nicht explizit gesichert (A9).

Es gibt bei der Erstellung von Artefakten keine Vorgaben zur angestrebten Abstraktionsebene (A10). Mögliche Überschneidungen der Inhalte von Artefakten sind nicht bekannt (A12). Zusätzlich werden gleiche Sachverhalte von verschiedenen Anforderungsingenieuren unterschiedlich modelliert (A13). Diese unterschiedliche Modellierung wirkt sich auf die Kommunikation aus und führt zu Missverständnissen (A14). Bei der Erstellung von Artefakten in frühen Phasen ist eine Vielzahl von Personen

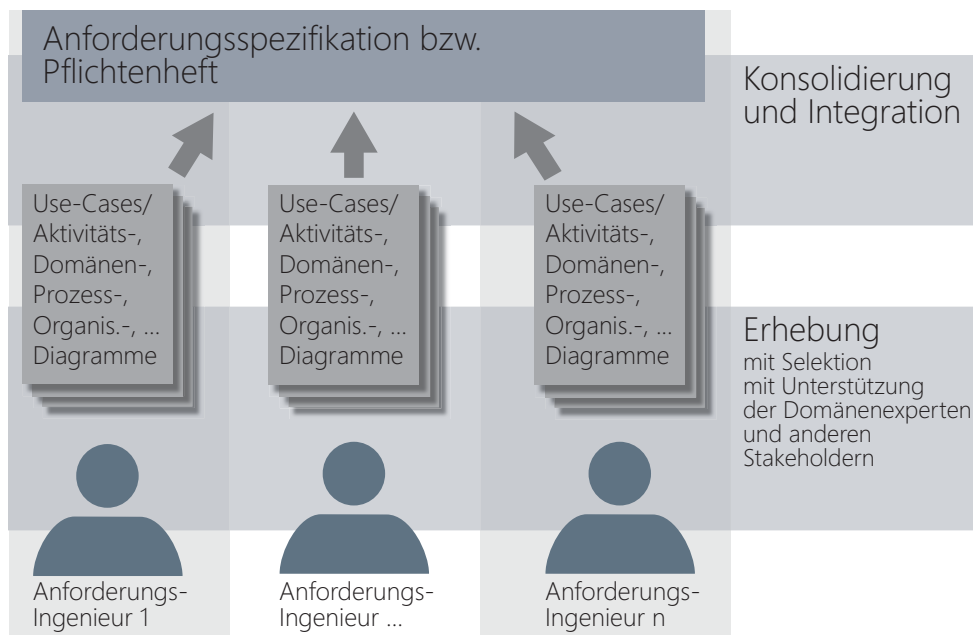


Abbildung 3.1: Zusammenfassung des Ergebnisses: Die Abbildung zeigt die durch die Erhebung in der Praxis gefundene Ausgangssituation im Requirements Engineering in frühen Phasen

beteiligt (B1 und B2). Dabei gibt es keine Artefakt-Verantwortlichkeiten (B4) und zusätzlich sind Artefakte nicht für alle Stakeholder verständlich oder sie widersprechen sich (B5).

Bei der Verwendung von Artefakten in späteren Phasen kommt es aufgrund von Mehrdeutigkeiten oder Widersprüchen häufig zu Problemen (z. B. müssen Details nachgefragt werden). Ursachen für die Mehrdeutigkeiten sind u.a. ungenau beschriebene Inhalte von Artefakten oder, dass die Artefakte voneinander losgelöst sind (C2).

Zusammenfassend, wie in Abbildung 3.1 zu sehen, sind im Requirements Engineering in frühen Phasen mehrere Personen beteiligt. Artefakte werden von diesen Personen typischerweise unabhängig voneinander erstellt. Dies geschieht mit Hilfe von Experten und anderen Stakeholdern, die Informationen zum realisierendem Softwaresystem haben. Am Ende dieser Schritte steht die Konsolidierung und Integration. Häufig besteht diese Integration aus einer reinen informalen Beschreibung eines Zusammenhangs zwischen den erstellten Artefakten. Die Integration findet in der Anforderungsspezifikation bzw. dem Pflichtenheft statt.

3.5.2 Forschungsfrage 3: Auftretende Probleme

Dieser Abschnitt beschreibt die aus den Ergebnissen extrahierten Probleme.

Inhaltliche Missverständnisse. Bei der Verwendung der Artefakt-Inhalte in frühen Phasen des Requirements Engineerings kommt es häufig zu Verständnisschwierigkeiten – z. B. sind die Artefakte nicht für alle Stakeholder lesbar (B5). Ursachen

dafür sind fehlende Vorgaben darüber, welches Wissen erfasst werden soll und wie die Erhebung stattfinden soll (A3).

Fehlendes Wissen wird nicht erfragt. Grund hierfür ist u.a. dass dies durch den Erheber auch nicht geschehen kann, da dieser nicht genügend Domänenwissen hat, um das fehlende Wissen zu identifizieren (A5).

Häufige Änderungen an Artefakten in frühen Phasen führen zu inhaltlichen Missverständnissen (A7). Ursache hierfür ist fehlende Methoden um Änderungen an die zuständigen Personen zu kommunizieren bzw. darauf aufbauende Artefakte entsprechend anzupassen.

Auswirkungen von Änderungen von Artefakt-Inhalten werden erst während der Integration der Artefakte in der Anforderungsspezifikation bzw. dem Lastenheft behandelt (A8). Artefakt-Inhalte in diesen Phasen sind häufig auch mehrdeutig und voneinander losgelöst (C2). Die Integration ist schwierig, da Wissen zur Integration fehlt oder auf Grund unterschiedlicher Sichten nicht integrierbar ist und/oder es keine Vorgaben zur Integration gibt (C7, C5 und C6).

Dies führt uns zur Aussage, dass es inhaltliche Missverständnisse gibt und dies ein Problem darstellen (siehe Abschnitt 5.2).

Unterschiedliche Abstraktionsebenen. Artefakte werden von unterschiedlichen Mitarbeitern erstellt, wobei die Sachverhalte von diesen Mitarbeitern unterschiedlich dargestellt werden (A13). Die Entscheidung, welches Wissen erhoben wird und wie es erhoben wird, unterliegt den einzelnen Mitarbeitern (A3).

Es entstehen Artefakt-Inhalte auf unterschiedlichen Abstraktionsebenen. Grund hierfür könnte u.a. sein, dass fehlendes Wissen nicht hinterfragt wird (A5). Die Nachbearbeitung von erhobenem Wissen in Artefakten wird großteils von einzelnen Personen durchgeführt (B2), dies führt dazu, dass fehlendes Wissen eher abstrakt erhoben wird und vorhandenes Wissen eher detailliert dargestellt wird.

Artefakt-Inhalte sind häufig mehrdeutig (C2), dies liegt u.a. daran, dass es keine Vorgaben zur gewählten Abstraktionsebene gibt (A10) und es kaum zu Überschneidungen in den Artefakt-Inhalten kommt (A12).

Dies führt uns zur Aussage, dass die unterschiedlichen Abstraktionsebenen von Artefakt-Inhalten ein Problem darstellen (siehe Abschnitt 5.3). Diese Aussage wird auch dadurch bestätigt, dass die Integration der Artefakt-Inhalte kaum möglich ist (C5).

Keine Konsistenzsicherung von Artefakten. Wie im Abschnitt *Unterschiedliche Abstraktionsebenen* eingeführt wurde, sind Artefakt-Inhalte häufig mehrdeutig (C2) und seitens der Unternehmen gibt es keine Vorgaben bzgl. der Abstraktionsebenen der Inhalte (A10). Die Inhalte werden großteils von den Mitarbeitern des Auftragnehmers ausgewählt (A3) und fehlendes Wissen kann nicht erhoben werden (A5). Dies führt dazu, dass die Artefakte nicht explizit konsistenzgesichert werden können (A9). Diese werden nur vom Ersteller inhaltlich überprüft (A9).

Konsistenzsicherung findet erst bei Integration in die Anforderungsspezifikation bzw. ins Lastenheft statt (A9). Die Inhalte sind aber häufig auch nicht für die Stakeholder verständlich (B5) und widersprechen sich dadurch auch nach der Integration

(B5, C7).

Dies führt uns zur Aussage, dass Artefakte bzw. deren Inhalte häufig inkonsistent sind und dies auch bleiben (B5) und dies ein weiteres Problem darstellt (u.a. bei der Integration – C7), siehe dazu Abschnitt 5.4.

Unterschiedliche Modellbildung durch Modellierer. Modelle werden von Modellierern unterschiedlich erstellt (A13). U.a. spielen in der Modellbildung subjektive Einflüsse eine Rolle, da es keine Vorgaben zur Auswahl des zu modellierenden Wissens gibt (A3), sowie die Abstraktionsebenen subjektiv bestimmt werden (A10). Dadurch gibt es bereits zwei Faktoren, die einen Einfluss auf die Erstellungsweise von Artefakten haben. Wie in der Analyse von Modellen ersichtlich (siehe dazu Abschnitt 5.5) werden gleiche Sachverhalte unterschiedlich dargestellt. Dies führt zu Kommunikationsproblemen, Missverständnissen und Integrationsproblemen (A14).

Wir leiten daher ab, dass die Modellbildung durch die Modellierer unterschiedlich geschieht und dies auch ein Problem darstellt, siehe dazu Abschnitt 5.5.

Unterschiedliche Prioritäten in der Modellbildung. Das erhobene Wissen wird vom Erheber seitens des Auftragnehmers bestimmt (A3) und zusätzlich von Domänenexperten getrieben (A3). Da es dazu keine Vorgaben gibt (A3), wird das z. B. durch den Domänenexperten in einem Interview preisgegebene Wissen durch den Erheber in einem Artefakt expliziert. Fehlendes Wissen kann aufgrund fehlendem Domänenwissens nicht erfragt werden (A5). Da die Artefakt-Inhalte sich in unterschiedlichen Artefakten kaum überschneiden (A12), werden in den Artefakten Teilbereiche erhoben, die nicht integrierbar sind (C5 und C7). Dies führt zu umfangreichen Problemen (C7).

Dies leitet uns zur Aussage, dass abhängig von den Domänenexperten das Wissen priorisiert erfasst wird und es daher zu einer unausgeglichenen Repräsentation von Wissen in Artefakten kommt, siehe dazu Abschnitt 5.6. Unausgeglichen ist die Repräsentation dann, wenn Ausschnitte aus der zu erfassenden Problem- und Lösungsdomäne in den Artefakten über- oder unterrepräsentiert erfasst sind.

Keine Verknüpfung von Artefakten zu anderen Artefakten. Bei der Integration kommt es zu einer Vielzahl von Problemen (C7), u.a. aufgrund von fehlendem übergreifendem Wissen. Übergreifendes Wissen zu repräsentieren, bedeutet verschiedene Sichten auf ein System zu integrieren, dies geschieht nicht, weil diese Sichten nicht integrierbar sind (C5). Hinzu kommt, dass die Artefakte nur in Ausnahmefällen integriert werden (C5). Die Auswirkungen dieser fehlenden Integration wurden hier bereits diskutiert.

Wir treffen die Aussage, dass es ein Problem ist, dass es keine bzw. nur eine geringfügige Verknüpfung von einem Artefakt zu anderen Artefakten gibt, siehe Abschnitt 5.7.

Ursachen für diese fehlende Integration sehen wir u.a. im Fehlen von Vorgaben zur Erhebung (A3).

Kein stakeholderweites Verständnis. Artefakt-Inhalte, erfasst in frühen Phasen des Requirements Engineerings, sind häufig mehrdeutig (C2) und inkonsistent (B5). U.a. liegt dies am fehlendem Verständnis über die Domäne (das sich z. B. durch das Fehlen von integrativen Wissen zeigt – C7, oder darin, dass auch nicht relevante Ausschnitte erfasst werden – C2). Ein weiteres Indiz dafür ist, dass Artefakt-Inhalte teilweise nicht verständlich und daher auch nicht einfach überprüfbar sind (B5). Dies liegt auch daran, dass die verwendete Notation nicht bekannt ist.

Obwohl die Artefakte in allen Phasen des Softwareentwicklungsprozesses eingesetzt werden (C1), müssen in späteren Phasen offene Missverständnisse geklärt werden (B5), was darauf hindeutet, dass es kein übergreifendes Verständnis gibt, auch nach dem Requirements Engineering nicht. Siehe dazu Abschnitt 5.8.

Problemzusammenhang. In Abbildung 3.2 haben wir den Zusammenhang zwischen den Problemen über die Antworten der Teilnehmer identifiziert. Die Pfeile repräsentieren eine Auswirkung auf ein anderes Problem.

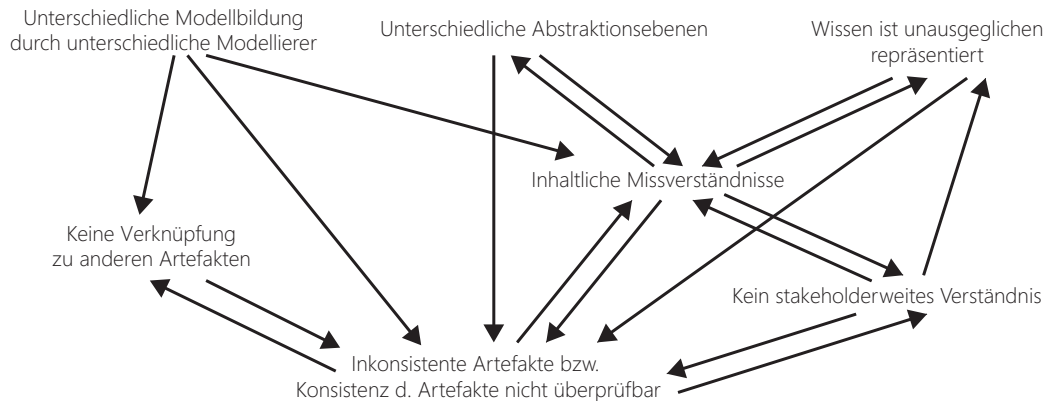


Abbildung 3.2: Problemzusammenhänge

3.6 Validität der Studie

Die größte Gefahr bezüglich Validität der Studie ist gegeben, durch die geringe Anzahl an befragten Personen. Die Studie kann daher nicht als repräsentativ angesehen werden.

Konstruktvalidität. Tiefeninterviews zeichnen sich besonders durch Erkenntnistiefe und der Qualität der Ergebnisse aus [KE10]. Die Umfrage wurde inhaltlich offen ausgeführt.

Im Gegensatz zu geschlossenen Fragen, konnten die Probanden freie Antworten geben. Das Missverstehen von Fragen konnte daher ausgeschlossen werden.

Kuss et al. [KE10] weisen neben den Vorteilen auch auf einige Probleme im Umgang mit Tiefeninterviews hin, auf die wir argumentativ kurz eingehen möchten.

- ▶ Zeitaufwand ist für Interviewer und Probanden hoch: Jedes der geführten Tiefeninterviews hatte eine Mindestlänge von 1,5 Stunden. Die Bereitschaft über das erwähnte Thema zu sprechen, unter der Bedingung, dass Anonymität gewahrt bleibt, war hoch und stellte für das Tiefeninterview kein Problem dar.
- ▶ Auskunftspersonen müssen gewissen intellektuellen Mindestanforderungen genügen: Die ausgewählten Personen wurden nach ihrer Tätigkeit in Softwareunternehmen und nach ihrer Erfahrung in Softwareprojekten ausgewählt. Dem Autor dieser Arbeit war daher sehr wohl bewusst, welcher Personenkreis befragt wird und, dass mit dieser Festlegung auch die Anwendbarkeit und Aussagekraft des Ansatzes auf den gewählten Personenkreis eingeschränkt ist.
- ▶ Ergebnisse müssen aufwändig aufbereitet werden: Die Mitschrift und daher Aufzeichnung der Tiefeninterviews erfolgte parallel zur Diskussion schriftlich vom Interviewer. Die Aufbereitung erfolgte, indem die Antworten der einzelnen Probanden, zu den gestellten Leitfragen, zusammengeführt wurden. Dies ist aufwändiger, als die Auswertung eines Fragebogens mit vorgefertigten Antworten, enthält aber qualitativ bessere Antworten und vor allem, werden die Antworten nicht durch die Antwortmöglichkeiten geleitet. Vor der Durchführung der Interviews war nicht vorhersagbar, wie die Ergebnisse aussehen werden, weshalb ein vorgefertigter Fragebogen mit Antwortmöglichkeiten daher auszuschließen war. Da in Tiefeninterviews auch die Argumentationsketten der Probanden mitartikuliert werden kann, erhielt man qualitativ hochwertige Antworten.

Durch den Aufbau der Interviews konnte vom Interviewer, wie auch von den befragten Personen, während der Beantwortung Fragen zur Klärung gestellt werden. Zusätzlich bestand für den Interviewer die Möglichkeit, innerhalb der Beantwortung einer Leitfrage, Fragen zur weiteren inhaltlichen Erschließung einer Antwort zu stellen.

Interne Validität. Die Probanden wurden auf den Kontext der Befragung hingewiesen. Alle Antworten bezogen sich auf frühe Phasen des Requirements Engineerings. Eine Studie durchgeführt von Neill und Laplante [NL03] kommt bei den Erhebungstechniken, trotz abweichender Branchen und der Tatsache, dass auch Personen an der Studie teilgenommen haben, die keine Softwareentwickler sind, zu einer ähnlichen Gewichtung der Erhebungs- und Erfassungstechniken. Andere Fragestellungen der von uns durchgeführten Studie adressiert die Studie von Neill und Laplante [NL03] nicht.

Partsch [Par10] weist darauf hin, dass 31% der Fehler in Anforderungen darauf zurückzuführen sind, dass Anforderungen fehlen oder inadäquat sind, 49% darauf, dass die Anforderungen inkorrekt und 18% darauf, dass sie inkonsistent oder unklar sind. Auch diese Aussagen decken sich mit der von uns durchgeführten Studie.

Externe Validität. Wohlin et al. [WRH⁺12] führt bei den Stichprobenverfahren u.a. auch die zweckmäßige Auswahl von Stichproben an. Dazu werden die Personen gewählt, die dem Zweck der Erhebung entsprechen. Unsere Auswahl von Probanden ist eine zweckmäßige Auswahl.

Wir sind uns bewusst, dass eine Studie mit der angeführten Anzahl an Teilnehmer keine statistische Relevanz hat. Davon abgesehen, sind aber in den Aussagen eindeutige Tendenzen zu erkennen, die die Existenz und Relevanz der gefundenen Probleme zeigt.

Die durchgeführten Interviews sind qualitative Interviews. Die Interviews wurden so lange durchgeführt, bis eine Informationssättigung eingetreten ist. Eine Informationssättigung tritt war, wenn zusätzliche Interviews keinen zusätzlichen Informationsgewinn mehr bringen.

Nach [WRH⁺12] ist die Anzahl einer Stichprobe immer dann zu erhöhen, wenn die Ergebnisse noch große Variabilität aufweisen. In den Aussagen haben wir keine größere Variabilität gefunden. Die Kernaussagen wurden mehrheitlich von einem Großteil der Teilnehmer getroffen.

Jeder der Befragten war bereits in mehreren Software-Projekten im vollen Umfang des Software-Entwicklungsprozesses beteiligt (vom Sammeln der Anforderungen bis hin zur Integration und Übergabe beim Auftraggeber) und kann eine mehrjährige Erfahrung (> 6 Jahre) als Softwareentwickler bzw. Anforderungs-Ingenieur aufweisen.

Es kann daher als gesichert gelten, dass die befragten Personen Experten in dem für diese Studie relevanten Bereich sind.

Die Größe der Entwicklerteams, in denen die befragten Personen tätig waren, schwankten zwischen 4 und 15 Entwickler. Ausgehend davon, dass Cockburn und Highsmith [CH01] erhoben haben, dass die durchschnittliche Teamgröße von agilen Softwareentwicklungsprojekten 9 Personen ist, sind die befragten Personen (wohl aber nicht die Anzahl) für agile Softwareentwicklungsprojekte repräsentativ.

Hinzu kommt, dass nach einer Erhebung von [FLW00] 86,9%⁴ aller datenverarbeitenden Unternehmen in den U.S.A. weniger oder gleich 20 Mitarbeiter haben. Ein Großteil der Unternehmen sind daher in der Größe der untersuchten Unternehmen bzw., durch die Aussage von Cockburn und Highsmith [CH01], deren Teamgröße.

Die Unternehmen der befragten Experten entwickeln Informationssysteme für sehr unterschiedliche Branchen (Groß- und Einzelhandel, allgemeinen Dienstleistungssektor, Industrieanlagen-Management, Sozialwirtschaft, ...).

In Softwareprojekten kann das initiale Problem auch vom Auftraggeber z. B. in Form eines Lastenheftes erarbeitet werden. In unserem Fall obliegt diese Aufgabe bereits dem Auftragnehmer.

Zuverlässigkeit. Die Gruppierung der Antworten innerhalb der Leitfragen durch den Interviewer unterliegt subjektiven Einflüssen. Die Antworten wurden stichwortartig transkribiert. Dabei wurde darauf geachtet, dass wenn eine Antwort nicht eindeutig (für den Interviewer) einer Gruppe zuzuordnen war, eine neue Antwortgruppe aus der Antwort gebildet wurde. Es ist nicht auszuschließen, dass eine andere Person, die die transkribierten Antworten ebenfalls Gruppen zuordnen soll, zu anderen Antwortgruppen kommt. Dies dürfte aber, angesichts dessen, dass Interviews, die ebenfalls an den Methoden und Problemen des Requirements Engineerings in

⁴ Nimmt man die bis 50 Mitarbeiter hinzu, sprechen wir bereits von 94,1%.

frühen Phasen interessiert waren (z.B. [NL03, FWLB10, Par10]), zu ähnlichen Ergebnissen kamen, keine Änderung der Problemtendenzen, die in dieser Arbeit gefunden wurden, nach sich ziehen.

Die Probanden sind Experten in ihrem Gebiet. Durch die geringe Anzahl der Befragten, kann es zu Abweichungen in den Aussagen von anderen Experten kommen. Die Tendenzen sollten sich aber nicht ändern. Dies kann auch dadurch bestätigt gesehen werden, dass Studien, die ähnliche Sachverhalte erheben, zu ähnlichen Ergebnissen gekommen sind, z.B. [NL03, FWLB10, Par10].

Analyse der in der Praxis in Artefakten erhobenen Inhalte

In diesem Kapitel stellen wir die Ergebnisse einer Studie vor, die im Rahmen der Beantwortung der Forschungsfrage 2 durchgeführt wurde. Ziel der Studie, war das Erheben der Inhalte von Artefakten, erstellt in frühen Phasen des Requirements Engineerings.

RQ2: Welche Inhalte werden in der Praxis in Artefakten erfasst, die zur Überbrückung der Kommunikationslücke in frühen Phasen des Requirements Engineerings erstellt wurden?

Zur Analyse wurden die im Rahmen von RQ1 befragten Personen gebeten, Artefakte zur Verfügung zu stellen, die in frühen Phasen des Requirements Engineerings und vor der Anforderungsspezifikation erstellt wurden.

Die extrahierten Inhalte liefern einen Beitrag zur Beantwortung von Forschungsfrage 3:

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

Die Beiträge zur Beantwortung der Forschungsfrage 3, werden in Kapitel 5 diskutiert, indem die Inhalte mit den Aussagen, die im Rahmen der Beantwortung von Forschungsfrage 1 und 3 in Kapitel 3 wiedergegeben wurden, in Bezug gesetzt werden.

Verschiedene Ausschnitte aus den analysierten Modellen werden in Kapitel 5 als Beispiel für das Auftreten von verschiedenen Problemen verwendet.

4.1 Kontext

Wir nehmen die Einteilung des Kontextes nach den Vorgaben von Wohlin et al. [WRH⁺12] vor.

Die Artefakte stammen von den Personen, die im Rahmen der Studie zur Beantwor-

tung der Forschungsfrage 1 teilgenommen haben.

Professionelles Umfeld. Die Teilnehmer der Studie wurden gebeten, typische Artefakte zur Verfügung zu stellen, die in frühen Phasen des Requirements Engineering, explizit vor Erstellung der Anforderungsspezifikation bzw. des Lastenheftes, erstellt wurden.

Reale Artefakte. Es wurden 33 Artefakte zur Verfügung gestellt. Alle Artefakte stammen aus Softwareprojekten, die in einem professionellen Kontext durchgeführt wurden.

Allgemeingültigkeit. Die Artefakte wurden von erfahrenen Softwareentwicklern bzw. Anforderungs-Ingenieuren erstellt. In der Literatur gibt es keine Vorgaben darüber, wie Artefakte in diesen Phasen zu erstellen sind. Es obliegt daher den Unternehmen, in denen die Personen tätig sind, welche Vorgehensweise sie wählen, um relevantes Wissen zu identifizieren und dieses in Artefakte zu explizieren.

4.2 Forschungsziel

Wir verwenden zur Formulierung des Forschungszieles die Schablone von Wohlin et al. [WRH⁺12].

- ▶ **Analysiert wird:** Artefakte erstellt in frühen Phasen des Requirements Engineering
- ▶ **Für den Zweck:** Evaluation der Hypothesen und Identifikation von Inhalten
- ▶ **Im Bezug auf:** Stand der Praxis
- ▶ **Aus Sicht von:** Softwareentwickler und Anforderungsingenieur
- ▶ **Im Kontext:** Artefakte des Requirements Engineering für die Entwicklung von prozessorientierten Informationssystemen

4.3 Forschungsfragen und Hypothesen

In der durchgeführten Studie wollen wir (a) Forschungsfrage 2 beantworten und (b) die auf Basis der durchgeführten Studie (siehe Kapitel 3) aufgestellten Hypothesen evaluieren.

H-1 Wissen über die Problem- und Lösungsdomäne werden vermischt. Artefakte in frühen Phasen des Requirements Engineerings enthalten bereits lösungsspezifische Informationen.

H-2 Gleiche Sachverhalte (z. B. Prozesse) werden von unterschiedlichen Personen unterschiedlich modelliert (z. B. benannte Eigenschaft, Entität).

Auf Basis der Ergebnisse aus Forschungsfrage 1, wollten wir in Modellen konkret erheben, wie sich die erwähnte unterschiedliche Repräsentation gleicher

Sachverhalte in Modellen wieder findet.

- H-3 Es werden nur wenige benannte Beziehungen in Artefakten verwendet. Beziehungen sind für das Verstehen einer Domäne von entscheidender Bedeutung. Wir gehen davon aus, dass Modelle nur geringfügig Beziehungen enthalten und dies mit ein Grund für die Schwierigkeiten bei Integration und Verständnis ist.
- H-4 Die Artefakte befinden sich auf unterschiedlichen Abstraktionsebenen. Die in frühen Phasen erstellten Artefakte werden häufig losgelöst voneinander erstellt. Abhängig vom Wissensträger¹ werden die erfassten Artefakte in unterschiedlichen Abstraktionsebenen erfasst.

4.4 Aufbau der Studie

Zur Beantwortung von Forschungsfrage 2 war es notwendig, die Inhalte der Artefakte zu analysieren. Dazu wurde eine Ontologie gesucht, deren Konzepte dazu verwendet werden konnten, die Inhalte der Modelle angemessen zu generalisieren und Gemeinsamkeiten zu identifizieren. Dazu wurde die Enterprise Ontologie von Uschold et al. [UKMZ98] verwendet. Die Konzepte in der Enterprise Ontologie stehen auch in Beziehung, diese Beziehungen wurden verwendet, um die Beziehungen zwischen Entitäten in den Artefakten zu explizieren.

Die zur Verfügung gestellten Artefakte wurden dann einzeln untersucht und die Inhalte gemäß der Enterprise Ontologie zugeordnet.

Die Modelle werden zusätzlich auf das Vorhandensein von lösungsspezifischen Informationen untersucht, um Hypothese H-1 evaluieren zu können. Wir definieren: Eine beliebige Information in einem Artefakt ist dann lösungsspezifisch, wenn das abgebildete Wissen nicht Bestandteil des abgebildeten Originals ist. Z. B. eine Identifikationsnummer, ID, zur eindeutigen Identifikation eines Objektes ist nicht Bestandteil des abgebildeten Originals oder aber auch, es gibt Entitäten im Modell, die nicht Bestandteil des abgebildeten Ausschnittes sind.

Zur Validierung der Hypothese H-2 analysieren wir, wie Aktivitäten in einem Modell repräsentiert sind. In Modellen die in frühen Phasen des Requirements Engineerings erstellt wurden ist es häufig notwendig Aktivitäten zu erfassen. Deshalb haben wir uns dazu entschlossen, Aktivitäten als Basis für die Untersuchung der Hypothese zu verwenden. Aktivitäten können in Modellen unterschiedlich repräsentiert sein. Wir gehen davon aus, dass die Möglichkeiten, wie Aktivitäten in Modellen repräsentiert sein können folgende sind: (a) als Entität, (b) als benannte Beziehung oder (c) als Funktions-Stub.

Zur Validierung der Hypothese H-3 wurden gezielt die benannten Beziehungen zur rein inhaltlichen Erhebung genauer betrachtet. Diese Beziehungen wurden den möglichen Beziehungen zwischen Konzepten der Enterprise Ontologie zugeordnet.

¹ Ein Wissensträger ist eine Person oder Gegenstand, die oder der als Quelle für das Erheben von Wissen dient.

4.5 Ergebnisse

Tabelle 4.2 zeigt die Ergebnisse der Analyse von Strukturmodellen aus frühen Phasen des Requirements Engineerings.

Jede Zeile entspricht einem analysierten Modell. Die Entitäten wurden den Konzepten der Enterprise Ontologie [UKMZ98] zugeordnet. Tabelle 4.1 zeigt die Menge von Entitäten mit verwendeten Abkürzungen. Die erste Spalte in der Tabelle 4.2 ist eine laufende Nummer, um die Modelle zu identifizieren. Die Spalte *Lsg.* gibt an, ob im Modell eindeutige Hinweise auf lösungsspezifische Inhalte gefunden wurden. Mehr zur Aussagekraft dieser Zahl wird in Abschnitt 5.2 erläutert.

Einordnung Enterprise Ontologie	Anzahl Entitäten
Voraussetzung (Cond.)	2
Spezifikation (Spec.)	6
Organizational Unit (Org. Unit)	38
Person	49
Ressource (Ress.)	208
Zustand einer Ressource (State)	22
Aktivität (Act.)	63

Tabelle 4.1: Ergebnis der Inhalte analysierter Strukturmodelle, Summen und Legende

Tabelle 4.1 zeigt die Anzahl der vorkommenden Entitäten, die den Konzepten der Enterprise Ontologie zugeordnet wurden. Analysiert wurden 33 Strukturmodelle. Vorgekommene Strukturmodelle waren: E/R-, Klassen- und Organisationsdiagramme.

Ergebnisse Beziehungen Tabelle 4.4 zeigt die Anzahl der Beziehungen in den analysierten Artefakten nach Beziehungs-Typ, gegeben durch den Syntax der Modellierungssprache.

Wir haben die Beziehungen in E/R-Diagrammen und den Organisationsdiagrammen denen der Klassendiagramme zugeordnet. Dies bedeutet, steht eine Entität mit einer anderen Entität in einer Beziehung *hat*, entspricht dies einer gerichteten Referenz. Wird dagegen z. B. *besteht (aus)* angegeben, wurde dies der Komposition zugerechnet. Alle weiteren textuellen Angaben wurden entsprechend ihrer Bedeutung aufgeteilt. Zusätzlich wurden alle benannten Beziehungen in Tabelle 4.6 zusammengefasst.

Benannte Beziehungen. In den Modellen kamen auch benannte Beziehungen vor, die über die rein durch den Syntax gegebenen hinaus gingen.

Die Beziehungen wurden, sofern dies möglich war, noch den Beziehungen zwischen Konzepten der Enterprise Ontologie zugeordnet.

Viele der Beziehungen waren nicht benannt und konnten daher nicht zugeordnet werden. Es wurden nur die Beziehungen der Enterprise Ontologie zugeordnet, die sich eindeutig zuordnen ließen, das bedeutet, dass wenn z. B. ein Funktions-Stub die Involviertheit eines Bearbeiters angedeutet hat (z. B. *VerarbeiteBearbeiter*), wurde dies

Nr.	Lsg.	Cond.	Spec.	Org. Unit	Person	Ress.	State	Act.	Sum.
1	n	0	0	4	4	4	2	0	14
2	n	0	0	1	1	4	1	1	8
3	j	0	0	3	1	6	0	4	14
4	j	0	0	0	1	8	5	2	16
5	n	0	0	0	0	3	5	0	8
6	n	0	0	1	2	10	1	3	17
7	j	0	4	0	1	10	0	2	17
8	j	0	0	2	3	2	0	1	8
9	n	0	0	2	2	6	0	0	10
10	j	0	0	1	0	3	0	1	5
11	n	0	0	10	4	2	0	2	18
12	j	1	0	0	1	5	0	2	9
13	j	0	0	0	3	4	0	1	8
14	j	0	2	1	0	4	2	4	13
15	j	0	0	0	1	15	0	0	16
16	j	0	0	1	2	4	0	1	8
17	j	0	0	0	0	14	0	2	16
18	j	0	0	0	1	8	0	1	10
19	n	0	0	0	3	16	0	0	19
20	j	0	0	1	3	3	1	1	9
21	j	1	0	1	2	1	0	2	7
22	n	0	0	0	0	8	0	0	8
23	n	0	0	0	1	9	4	0	14
24	j	0	0	0	1	14	0	0	15
25	n	0	0	2	4	5	0	8	19
26	j	0	0	1	0	6	0	1	8
27	j	0	0	0	0	10	0	3	13
28	j	0	0	2	3	3	1	1	10
29	j	0	0	1	0	3	0	6	10
30	n	0	0	3	2	8	0	4	17
31	j	0	0	1	2	4	0	0	7
32	j	0	0	0	0	6	0	3	9
33	n	0	0	0	1	0	0	7	8
Sum.	21 (j)	2	6	38	49	208	22	63	

Tabelle 4.2: Detaildarstellung der Inhalte analysierter Strukturmodelle

Art der Beziehung	Anzahl
Komposition (Komp.)	36
Aggregation (Aggr.)	47
Gerichtete Referenz (ger. Ref.)	144
Ungerichtete Referenz (unger. Ref.)	142
Spezialisierung (Spez.)	86

Tabelle 4.3: Ergebnis der Inhalte bezüglich Beziehungen, Summen und Legende

Nr.	Komp.	Aggr.	ger. Ref.	ung. Ref.	Spez.	Sum.
1	6	0	5	2	5	18
2	1	1	2	5	0	9
3	0	1	7	4	2	14
4	0	0	12	8	3	23
5	0	0	7	0	0	7
6	0	1	3	9	3	16
7	0	0	7	6	18	31
8	0	0	12	0	0	12
9	0	0	0	9	0	9
10	0	1	0	4	0	5
11	7	4	0	11	0	22
12	0	0	12	0	0	12
13	5	0	0	5	2	12
14	0	7	3	0	0	10
15	0	0	3	0	15	18
16	1	0	0	5	2	8
17	0	5	0	2	9	16
18	0	0	11	0	0	11
19	7	0	0	17	0	24
20	0	0	0	8	0	8
21	0	0	6	0	0	6
22	0	8	0	0	2	10
23	0	7	0	1	7	15
24	2	0	0	10	14	26
25	0	0	0	17	1	18
26	0	5	3	0	0	8
27	0	7	8	0	0	15
28	0	0	11	0	0	11
29	0	0	10	0	0	10
30	0	0	0	19	0	19
31	0	0	10	0	0	10
32	7	0	0	0	3	10
33	0	0	12	0	0	12
Sum.	36	47	144	142	86	

Tabelle 4.4: Detaildarstellung Ergebnis Inhalte bezüglich Beziehungen

nicht als Beziehung zum Bearbeiter identifiziert.

Da die Enterprise Ontologie [UKMZ98] keine weitere Unterteilung in der Involviertheit einer Ressource in eine Aktivität zulässt als die allgemein Involviertheit, haben wir uns entschlossen, diese Einteilung durch die von Dori [Dor11] zu erweitern.

Art der Beziehung	Anzahl
Ressource lesend (R. les.)	6
Ressource ändernd (R. änd.)	10
Ressource erstellend (R. erst.)	8
Ressource löschend (R. lösch.)	2
<i>Performer</i> , ausführende Rolle (<i>perf.</i>)	12
Enthält	14
Referenziert (<i>ref.</i>)	21
Spezialisiert (<i>spez.</i>)	3
Spezifikation einer Aktivität (<i>A. spec.</i>)	3

Tabelle 4.5: Ergebnis Inhalte bezüglich benannter Beziehungen, Summen und Legende

Der Informationsgewinn der Beziehungen *enthält*, *referenziert* und *spezialisiert* geht nicht über das hinaus, was bereits durch die Analyse des durch den Syntax gegebenen Beziehungen ersichtlich war und ist nur der Vollständigkeit halber angegeben, da diese Beziehungen in den Modellen explizit benannt wurden.

Repräsentation von Aktivitäten. Tabelle 4.7 zeigt die Ergebnisse der Analyse, wie eine Aktivität in den Modellen repräsentiert ist.

Aktivitäten sind in 73% der Modelle vorhanden.

Häufig werden Aktivitäten über Beziehungen ausgedrückt (z. B. *erstellt*), dies geschieht in 12 Modellen (50%), in sieben Modellen werden die Aktivitäten durch die explizite Benennung in Entitäten dargestellt (ca. 29%) und in nur fünf Fällen werden dazu Funktions-Stubs verwendet (ca. 21%).

In zwei Modellen werden Beziehungen und explizite Benennung von Entitäten verwendet und in drei Modellen werden Funktions-Stubs und Beziehungen verwendet. Diese Zahlen sind bereits in den Einzelrepräsentation enthalten.

Beziehungen zwischen Artefakten. Wir haben in den Ergebnissen keine Analyse der Beziehungen zwischen Artefakten präsentiert. Wir waren initial an den Eigenschaften der Beziehungen zwischen Artefakten interessiert, diese werden in Kapitel 3 erhoben und in Kapitel 5 detailliert.

4.6 Analyse und Interpretation

Beantwortung der Forschungsfrage

Nr.	R. les.	R. änd.	R. erst.	R. lösch.	perf.	enth.	ref.	spez.	A. spec.	Sum.
1	1	0	0	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	0	1
3	3	1	2	0	0	1	0	0	0	7
4	1	1	0	0	0	0	0	0	1	3
5	0	0	0	0	0	0	0	0	0	0
6	0	1	0	0	0	1	1	1	0	4
7	0	0	1	1	1	2	5	1	0	11
8	0	0	0	0	2	0	0	0	0	2
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0
13	0	1	0	1	0	1	0	0	0	3
14	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	1	0	0	0	1
16	1	0	0	0	0	0	0	0	0	1
17	0	1	0	0	0	0	0	0	1	2
18	0	1	1	0	0	0	1	0	0	3
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	2	1	0	0	3
21	0	0	2	0	0	1	0	0	0	3
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	2	0	0	1	3
25	0	0	0	0	0	1	3	0	0	4
26	0	0	0	0	0	1	3	0	0	4
27	0	0	0	0	0	0	4	0	0	4
28	0	2	1	0	0	0	3	0	0	6
29	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	6	0	0	0	0	6
31	0	1	1	0	3	0	0	0	0	5
32	0	0	0	0	0	1	0	1	0	2
33	0	0	0	0	0	0	0	0	0	0
Sum.	6	10	8	2	12	14	21	3	3	

Tabelle 4.6: Detaildarstellung Inhalte bezüglich benannter Beziehungen

Art der Repräsentation	Anzahl d. Modelle mit Vorkommen
Beziehung	12
Entität	7
Funktions-Stub	5
Beziehung und Entitäten	2
Funktions-Stub und Beziehungen	3

Tabelle 4.7: Ergebnisse Repräsentation von Aktivitäten

RQ2: Welche Inhalte werden in der Praxis in Artefakten erfasst, die zur Überbrückung der Kommunikationslücke in frühen Phasen des Requirements Engineerings erstellt wurden?

Die erhobenen Inhalte werden in den Ergebnissen in Abschnitt 4.5 detailliert dargestellt.

Zur Teilbeantwortung der Forschungsfrage 3

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

wollen wir die auf Basis von Forschungsfrage 1 erstellten Hypothesen erläutern.

H-1 Wissen über die Problem- und Lösungsdomäne werden vermischt. Artefakte in frühen Phasen des Requirements Engineerings enthalten bereits lösungsspezifische Informationen. Ca. 64% der Modelle enthalten eindeutig lösungsspezifisches Wissen.

Lösungsspezifisch ist explizites Wissen dann, wenn es nicht Bestandteil des abgebildeten Originals ist. Z. B. eine Identifikationsnummer, ID, zur eindeutigen Identifikation eines Objektes ist nicht Bestandteil des abgebildeten Originals oder aber auch, es gibt Entitäten im Modell, die nicht Bestandteil des abgebildeten Ausschnittes sind.

Diese Zahl bestätigt unsere Hypothese.

H-2 Gleiche Sachverhalte (z. B. Prozesse) werden von unterschiedlichen Personen unterschiedlich modelliert (z. B. benannte Eigenschaft, Entität). Wir haben unter Verwendung eines häufig vorkommenden Teilproblems (der Darstellung einer Aktivität bzw. eines Vorganges) versucht herauszufinden, wie viele unterschiedliche Darstellungen vorkommen.

Aktivitäten kamen in 73% der Modelle vor. Aktivitäten werden in 50% der Fälle als Beziehung dargestellt, in ca. 29% der Fällen als Entitäten und in ca. 5% der Fälle als Funktions-Stub. In ca. 21% der Modelle kamen jeweils zwei der genannten Fälle innerhalb eines Modelles vor.

Wir sehen die Hypothese als bestätigt. Darüber hinaus werden Aktivitäten sogar innerhalb von Modellen bereits unterschiedlich modelliert.

H-3 Es werden nur wenige benannte Beziehungen in Artefakten verwendet. In den analysierten Modellen wurden 455 Beziehungen gefunden, davon waren 79 explizit benannt. Das entspricht ca. 17 %. Zieht man von dieser Zahl noch die Beziehungen ab, die bereits über den Syntax der Modellierungssprache ausgedrückt wurden (Komposition, Spezialisierung, und reine Referenzierung in Summe 38 Beziehungen), verbleiben 41 Beziehungen, die eine inhaltliche Aussagekraft über die beschriebene Domäne haben. Das entspricht ca. 9 %.

Diese Zahlen bestätigen unsere Hypothese.

H-4 Die Artefakte sind auf unterschiedlichen Abstraktionsebenen. Die Abstraktionsebenen der Modelle sind nur qualitativ messbar. Wir haben dazu in der Analyse in Abschnitt 5.3 Beispiele aus der Praxis angeführt, die diese Hypothese untermauern.

Weiterer Beitrag zur Identifikation von Problemen. In Kapitel 3 wurden weitere Probleme identifiziert, die mit Hilfe der Modelle überprüft werden können. Dies geschah in diesem Abschnitt durch die Hypothesen und wird in Kapitel 5 durch Ausschnitte aus den Modellen nochmals untermauert.

Beziehungen zwischen den Artefakten. Es wurde im Rahmen der Analyse der Inhalte der Artefakte aus gleichen Softwareprojekten auch versucht, die Artefakte in eine Beziehung untereinander zu setzen.

Eine Integration der Inhalte war nur bei wenigen Inhalten in Artefakten möglich. Einerseits wurden Entitäten teilweise unterschiedlich benannt und eine eindeutige Zuordnung zu einem anderen Artefakt war nicht möglich. Andererseits hat sich auch bestätigt, was in Kapitel 3 bereits von den Probanden erwähnt wurde, nämlich, dass die Artefakte auf einer unterschiedlichen Abstraktionsebene sind und auch deshalb ein Zusammenhang nicht einfach identifiziert werden kann, ohne eine Interpretationsschlücke schließen zu müssen.

Wir haben festgestellt, dass bei der Integration von Artefaktinhalten tiefes Wissen über die Problemdomäne vorhanden sein muss, da ansonsten das fehlende Wissen durch Nachfrage, oder im schlimmsten Fall durch Interpretation, geschlossen werden muss. Dies erklärt auch, warum in der in Kapitel 3 präsentierten Studie von den Probanden erwähnt wurde, dass die Integration der Artefaktinhalte häufig nur informal beschrieben wird und die Integration der Inhalte daher eigentlich überhaupt nicht stattfindet.

4.7 Validität der Studie

Konstruktvalidität. Es ist möglich, dass unsere Einteilung in die Konzepte der Enterprise Ontologie für die dargestellten Inhalte nicht angebracht waren. Die Enterprise Ontologie [UKMZ98] wurde vielfach in der Praxis verwendet, um Unternehmen in Aufbau und Ablauf zu beschreiben. Sie stellt daher eine vielbenutzte Basis dar. Coughlan und Macredie [CM02] heben hervor, dass das Verstehen eines Problemereiches aus dem Verstehen der Benutzer, dem Arbeitsbereich und den allgemeinen Organisationsstruktur besteht. Die Enterprise Ontologie deckt diese Bereiche (und viele darüber hinaus) ab, und schien für den Zweck der inhaltlichen Analyse deshalb auch geeignet.

Interne Validität. Die Auswahl der Artefakte konnte nicht beeinflusst werden. Es wurden alle übermittelten Artefakte analysiert. Es ist möglich, dass die übermittelten Artefakte nicht repräsentativ sind. Die Teilnehmer wurden gebeten, Artefakte zu übermitteln, die typischerweise in frühen Phasen des Requirements Engineerings erstellt wurden.

Es wurden nur Strukturdiagramme zur Analyse übermittelt. Obwohl in der Studie angegeben wurde, dass in frühen Phasen des Requirements Engineering noch Artefakte anderer Artefakt-Typen erstellt werden. In den Inhalten der übermittelten Artefakte sind Tendenzen zu erkennen, die unsere Problemstellung unterstützen.

Externe Validität. In der Analyse wurden alle Entitäten mit ihren Beziehungen Schritt für Schritt betrachtet und deren *gemeinter* Bedeutung identifiziert und die verschiedenen im Design angesprochenen Analysen durchgeführt. Bei der Identifikation der Bedeutung der benannten Beziehungen ist die *gemeinte* Bedeutung nicht immer einfach zu identifizieren. Auf Basis der anderen Entitäten innerhalb eines Artefakts haben wir versucht, den Zusammenhang zu identifizieren. Es bleibt das Risiko, dass diese Zuordnung nicht vollständig korrekt möglich war. Dies sollte aber die Aussagen der Studie nicht verändern. Selbst wenn die benannten Beziehungen nicht vollständig richtig zugeordnet wurden, machen sie im Gesamten nur ca. 17% aller Beziehungen in den Modellen aus.

Zuverlässigkeit. Wie oben stehend erwähnt, ist die Einteilung der Beziehungen nach ihrer *gemeinten* Bedeutung nicht einfach und subjektiv. Diese Beziehungen machen aber nur einen Bruchteil der Beziehungen in den Artefakten aus. Daher wirkt sich eine Abweichung hier nur zu einem kleinen Teil auf die Tendenzen der gefundenen Inhalte aus.

Interpretation und Ergebnisse, Ableitung einer Lösungsidee

In der Problemanalyse werden Probleme benannt und erläutert, die im Rahmen der Ergebnisse der Beantwortung von Forschungsfrage 3 aufgetreten sind.

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

Für Forschungsfrage 3 ist die Beantwortung der Forschungsfragen RQ1 und RQ2 notwendig, diese wurden in Kapitel 3 und Kapitel 4 behandelt.

Abbildung 5.1 zeigt, dass die Problemanalyse auf den Forschungsfragen 1 und 2 basiert. Dieses Kapitel fasst die in Kapitel 3 und 4 durchgeführten Analysen zusammen. Basierend auf den Ergebnissen führen wir eine Interpretation durch und leiten daraus eine Lösungsidee ab.



Abbildung 5.1: Die Problemanalyse fasst die Ergebnisse aus Forschungsfrage 1 und 2 zusammen, die Pfeile symbolisieren den Ablauf der Analyse

Abbildung 5.2 zeigt die identifizierten Probleme und deren direkt gefundenen Zusammenhang. Wir beschreiben in diesem Kapitel die einzelnen Probleme, von deren Entstehung, über deren Auswirkungen und einigen Beispielen bis zu der Lösungsidee, die in dieser Arbeit verfolgt wird.

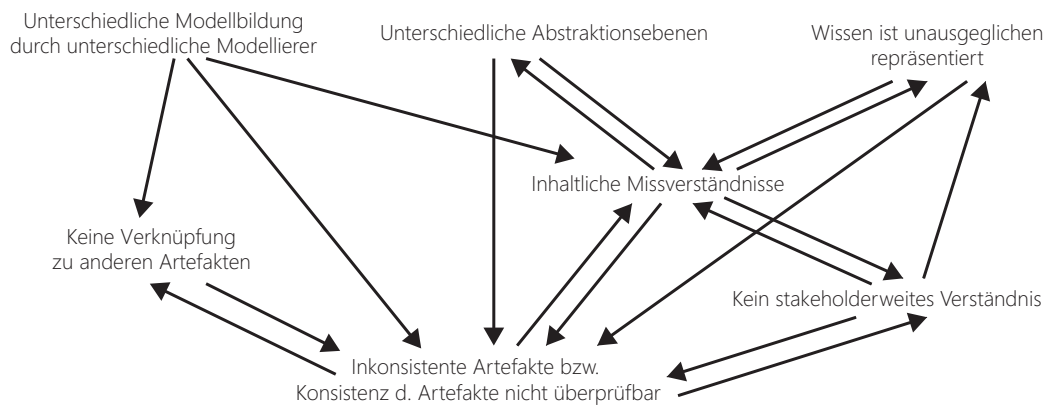


Abbildung 5.2: Problemzusammenhänge

5.1 Darstellung der analysierten Probleme

Jedes Problem wird wie folgt beschrieben:

- ▶ **Entstehung und Ursachen des Problems:** Dieser Abschnitt beschreibt die Problem Entstehung bzw. verschiedene Ursachen des Problems.
- ▶ **Beispiel:** Dieser Abschnitt enthält zu einem Problem ein oder mehrere Beispiele (sofern sinnvoll), die das Problem charakterisieren.
- ▶ **Auswirkungen:** Dieser Abschnitt beschreibt die Auswirkungen des Problems.
- ▶ **Lösungszustand und Lösungsidee:** Beschreibt (a) einen Zustand, der das Problem lösen würde und (b) die Lösungsidee, die im Rahmen dieser Arbeit verfolgt wird, um zur Lösung des Problems beizutragen.
- ▶ **Kriterium:** Enthält ein Kriterium, das einen gelösten Zustand beschreibt.

Die Kriterien werden wie folgt nummeriert:

K99.1: K steht für Kriterium, 99 für die Problemnummer und 1 ist das erste Kriterium des Problems.

Wir verwenden diese Kriterien, wie beschrieben in [Bro02], um die Merkmale zu benennen, mit deren Hilfe die Erfüllung von Sachverhalten auf wahr oder falsch unterschieden werden kann.

5.2 Problem 1: Inhaltliche Missverständnisse

In Artefakten, erstellt in frühen Phasen des Requirements Engineerings, wird unterschiedliches Wissen erfasst. Durch das Fehlen von Vorgaben zur Erhebung und unterschiedlichen Abstraktionsebenen des erfassten Wissens, kommt es zu inhaltlichen Missverständnissen.

5.2.1 Entstehung bzw. Ursache des Problems

Inhaltliche Missverständnisse entstehen aus vielen Gründen, u.a. spielt die unterschiedliche Modellbildung durch die Modellierer, die Tatsache, dass die Artefakt-Inhalte unterschiedliche Abstraktionsebenen aufweisen, die unausgeglichene Repräsentation von Wissen und die Inkonsistenz von Artefakten untereinander eine Rolle dabei.

Weitere Gründe für inhaltliche Missverständnisse sind häufige Änderungen in Artefakten in frühen Phasen. Durch das Fehlen einer Integration der Artefakte nach der Erhebung, sind diese Artefakte häufig losgelöst voneinander. Änderungen in einem Artefakt haben daher keine Auswirkungen auf andere Artefakte. Dem wird in der Praxis damit begegnet, dass die Artefakte in die Anforderungsspezifikation bzw. dem Lastenheft integriert werden und dort eine inhaltliche Überprüfung stattfindet. Das Ergebnis unserer Befragung zeigt, dass diese Integration das Problem nicht löst, da der Zusammenhang zwischen Artefakten häufig nur textuell beschrieben wird und Lücken bleiben, die dann während der Entwicklung überbrückt werden müssen z. B. durch Interpretation durch den Entwickler. Dass dies geschieht, bestätigt die durchgeführte Studie.

Diese angesprochenen Probleme werden im Detail in den weiteren Abschnitten dieses Kapitels behandelt. In diesem Abschnitt beschreiben wir inhaltliche Probleme, die sich aufgrund der modellierten Sachverhalte in Modellen in der Praxis ergeben können. Die Aussagen basieren auf die durchgeführte Studie in Kapitel 4.

Inhaltliche Missverständnisse aufgrund von Modellinhalten. Wir haben festgestellt, dass in Strukturdiagrammen, erstellt in frühen Phasen des Requirements Engineering, 64% der Modelle eindeutig lösungsspezifische Informationen¹ enthalten. Dies passt zu der Aussage aus unserer Studie, dass die Erhebung vom Erheber getrieben wird – in Abhängigkeit vom Domänenexperten, der die Inhalte prägt. Die Darstellung der erhobenen Sachverhalte obliegt dem Erheber und dazu gibt es keine Vorgabe seitens der Unternehmen.

Auch Partsch [Par10] stellte fest, dass im Requirements Engineering oft schon eindeutige Vorgaben oder Lösungskonzepte für den Entwurf und manchmal sogar auch für die Implementierung festgeschrieben werden.

Durch die vorhin beschriebene Aussage, dass Artefakte in frühen Phasen bereits lösungsspezifische Informationen enthalten, ist auch gegeben, dass dies zu zusätzlichen Verständnisschwierigkeiten führt, da die Software-Artefakte häufig nicht verstanden werden (siehe dazu durchgeführte Studie) und Informationen enthalten, die nicht im Original (dem abgebildeten Sachverhalt) vorhanden sind. Aus denselben Gründen ist auch für die Domänenexperten ein Review der Artefakte nicht einfach durchführbar.

Ein weiteres Problem, das in der Praxis dadurch auftritt sind Missverständnisse, die darauf zurückzuführen sind, dass auch Entitäten in diesen Modellen vorhanden

¹ Lösungsspezifisch ist eine Information dann, wenn sie nicht Bestandteil des abgebildeten Originals ist. Z. B. eine Identifikationsnummer, ID, zur eindeutigen Identifikation eines Objektes ist nicht Bestandteil des abgebildeten Originals oder aber auch, es gibt Entitäten im Modell, die nicht Bestandteil des abgebildeten Ausschnittes sind.

sind, die später nicht in der Software zu finden sind.

In Kapitel 3 wurden noch weitere Entstehungsgründe eingeführt.

In der Literatur. Nach Partsch [Par10] liegen Ursachen für gescheiterte Softwareprojekte häufig in einer nicht verstandenen Problemstellung, unzureichend formulierten Anforderungen und unklaren Zielformulierungen.

5.2.2 Beispiel

In Abschnitt 3 wurde angegeben, dass bei bestimmten Artefakten, die in der Anforderungsspezifikation integriert wurden, Probleme bezüglich den Inhalte auftreten. Z. B. wegen Entitäten, die nicht in der Software abgebildet werden sollen, aber bereits, beispielsweise, als Datenbank-Entität in der Lösung vorhanden sind.



Abbildung 5.3: Ausschnitt eines Modells: Beispiel für lösungsspezifischer Inhalt in Modell

Abbildung 5.3 zeigt ein aus der Praxis stammendes Beispiel (Ausschnitt). Die Abbildung zeigt zumindest eine lösungsspezifische Entität, gekennzeichnet mit dem Postfix *Impl*. Zusätzlich wird auch eine Methode angegeben, die den Namen *ProcessLoanFunding* trägt. Die Methode kann noch als Prozess betrachtet werden, der unter Einbeziehung einer Entität ausgeführt werden kann.

Der Postfix zeigt aber eindeutig, dass in diesem Fall das Modell lösungsspezifische Informationen enthält. Das Problem ist, dass die anderen Entitäten durchaus rein problemspezifisch sein können. Hier gibt es keinen Hinweis darauf, ob es sich um die Abbildungen in der Software handelt (wobei dies in Anbetracht der Verwendung des Postfix *Impl* bei der involvierten Entität gegeben sein wird). Besonders zu beachten ist in diesem Beispiel, dass es einerseits eindeutige Informationen enthält, die lösungsspezifisch sind, aber dennoch versucht, die Problemdomäne zu beschreiben.

Abbildung 5.4 zeigt einen Ausschnitt eines Modells aus der Praxis. Zu sehen ist, dass das Modell lösungs- und problemspezifische Inhalte mischt. Z. B. wird in der Entität *Employee* eine *employeeID* und ein *password* angeführt. Beides ist nicht Bestandteil des Originals, sondern würde sich erst durch die Anforderungen an das Softwaresystem ergeben.

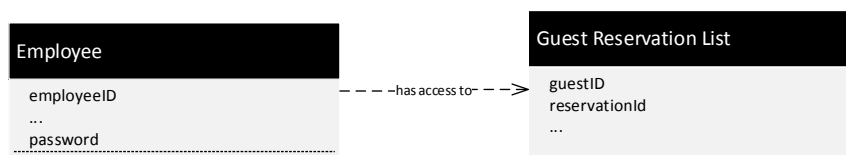


Abbildung 5.4: Ausschnitt eines Modells: Beispiel für missverständlichen Inhalt

Besonders interessant ist die Beziehung *has access to*, die ausdrückt, dass ein *Employee* auf eine bestimmte Liste Zugriff hat. Einerseits gibt dieses Beispiel vor, wie eine kon-

krete Lösung zum dargestellten Sachverhalt aussieht und andererseits nimmt das Modell durch diese Beziehung davon Abstand nimmt.

5.2.3 Auswirkungen

Die in Abschnitt 5.2.2 eingeführten Beispiele zeigen Modelle aus der Praxis, die Inhalte enthalten, die nicht Teil der Anforderungserhebung sind und zu inhaltlichen Missverständnissen führen.

Für inhaltliche Missverständnisse gibt es viele Gründe, u.a. bereits in Abschnitt 5.2.1 und in der Analyse der Studie, in Kapitel 3 erläutert.

Die Auswirkungen von inhaltlichen Missverständnissen sind vielfältig. Z. B. inkonsistente Artefakte werden in die Anforderungsspezifikation bzw. dem Lastenheft integriert und dienen als Basis für die weiteren Phasen des Softwareentwicklungsprozesses. Solche Missverständnisse, die zu inkonsistenten Anforderungen führen können im schlimmsten Fall erst bei Integration des fertigen Systems in die Anwendungsdomäne entdeckt werden, da die Anforderungen durchaus valide sein können, auch wenn sie inkonsistent sind.

5.2.4 Mögliche Lösung des Problems

Ein möglicher Beitrag zur Problemlösung kann darin bestehen alle modellierten Entitäten zu stereotypisieren. Die Rolle der modellierten Entitäten soll in Bezug auf Problem- und Lösungsdomäne ersichtlich sein. Dies bedeutet, wird eine Entität in der Lösungsdomäne erfasst, aus der eine Entität eines Datenmodells gebildet wird und stellt diese eine Abbildung eines in der Problemdomäne existierenden Originals dar, soll ersichtlich sein, dass es sich um eine lösungsdomänenspezifische Abbildung handelt und nicht um eine Abbildung des problemdomänenspezifischen Originals.

Um festzustellen, welche Eigenschaften eine Entität aufweisen muss, muss auch ersichtlich sein, welcher Ausschnitt aus der Problemdomäne abgebildet wird. Die Beziehungen zwischen Lösungs- und Problemdomäne müssen daher ersichtlich sein.

Lösungsidee. Da verschiedene Modellierungstechniken in frühen Phasen des Requirements Engineering eingesetzt werden und Stereotypisierung kein Konzept ist, welches bei allen gegeben ist, verfolgen wir in dieser Arbeit eine andere Idee. Wir bieten ein Begriffsmodell, das wir Basismodell nennen, und als Terminologie für alle in frühen Phasen zu erstellende Artefakte dient. Dies bedeutet konkret, dass die Entitäten, die in Artefakten verwendet werden, aus diesem Basismodell stammen. Die im Basismodell erfassten Begriffe sind in einem Erhebungsvorgang erfasst worden und bereits stereotypisiert. Die Begriffe stehen in Beziehung zueinander, diese Beziehungen sind ebenso stereotypisiert. Die Begriffe stammen aus der Problem- und Lösungsdomäne. Bei der Modellierung des z. B. in Abbildung 5.3 zu sehenden Sachverhaltes wären die Entitäten einem Begriff im Basismodell zuordenbar.

Beispiel zur Lösungsidee. Das in Abbildung 5.4 dargestellte Beispiel enthält bereits lösungsspezifische Informationen in Form von Identifikationsnummern. Dabei

weist die Beziehung *has access to* zwischen den Entitäten auf die konzeptuelle Natur des Modells hin.

Zur Lösung dieses Problems versuchen wir die zwei in frühen Phasen des Entwicklungsprozesses vorkommenden Domänen in einem Basismodell zu verbinden. Abbildung 5.5 zeigt ein solches Basismodell. Alle Entitäten und Beziehungen darin sind stereotypisiert.

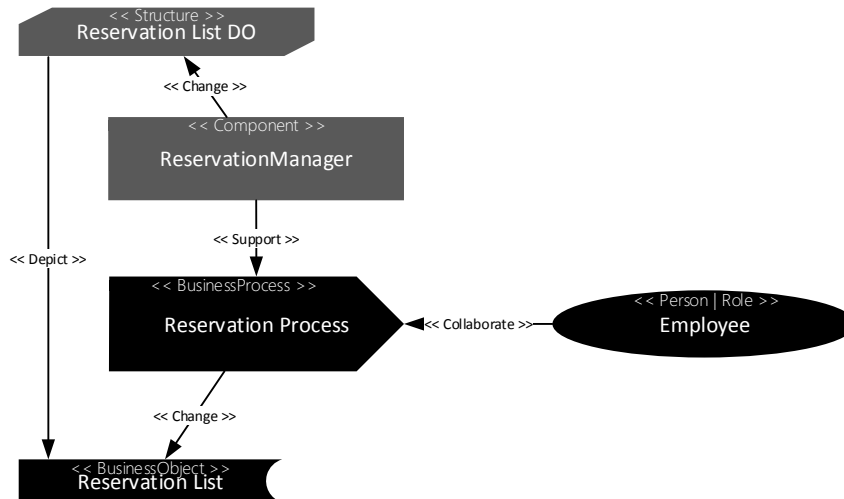


Abbildung 5.5: Lösung des Problems in Abbildung 5.4

Es zeigt u.a. dass es eine Komponente *ReservationManager* gibt, die den Geschäftsprozess *Reservation Process* unterstützt. Soll nun ein Ausschnitt detailliert werden, dienen diese Entitäten aus dem Basismodell als Bezeichner für die Entitäten in anderen Artefakten. Nehmen wir an, wir möchten die *has access to* aus der Abbildung 5.4 detaillieren. Voraussetzung dafür ist es zu verstehen, was dies bedeutet. Der Zugriff kann bedeuten, dass der *Employee* reinen Lesezugriff auf die *Guest Reservation List* hat, es kann aber auch bedeuten, dass der *Employee* für die Reservierung zuständig ist, und die Liste auch ändert.

Wir nehmen an, dass wir die Beziehungen geklärt haben und, dass das durch das Modell in Abbildung 5.4 offensichtlich entstehende angesprochene Missverständnis bereits durch die Einführung der Stereotypisierung von Beziehungen reduziert werden kann.

Auf Basis des Basismodells in Abbildung 5.5 können wir nun beginnen Ausschnitte zu Detaillieren, z. B. die Konstellation *ReservationManager* und *Employee* mit einem Sequenzdiagramm. Dabei muss die Veränderung des *Reservation List DO* berücksichtigt werden. Dies klärt die vage Aussage *has access to*.

Weiter könnte es der Fall sein, dass der Modellierer des Modells in Abbildung 5.4 ein Datenmodell erstellen wollte. Hierzu kann das Basismodell ebenso herangezogen werden. Die mit den Stereotypen *Struktur* versehenen Entitäten entsprechen Entitäten in Datenmodellen. Dies bedeutet, das Basismodell trennt zwischen der Lösungs- und Problem-domäne indem es u.a. die Beziehungen zwischen Entitäten aus der Lösungs-domäne und der Problem-domäne explizit macht.

Dies ist besonders wichtig, da Eigenschaften von Abbildungen in der Problem-domä-

ne, von denen in der Lösungsdomäne häufig abweichen.

5.2.5 Kriterium

Modelle werden vom Menschen interpretiert. Inhaltliche Missverständnisse in der Art wie sie in diesem Abschnitt erläutert werden, sind dann unwahrscheinlicher, wenn die Leser der Modelle zu einer ähnlichen Interpretation über die Inhalte kommen.

Wir leiten daraus das Kriterium ab:

- ▶ **K1.1:** Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.

Etwas ist dann mit etwas anderem ähnlich, wenn es dieses in bestimmten Merkmalen übereinstimmt [Dud11]. Wir legen fest, dass eine Interpretation eines Modells ähnlich ist, wenn die Bedeutung der in natürlicher Sprache artikulierten Inhalte über die Entitäten und Beziehungen in einem Modell personenübergreifend übereinstimmt.

5.3 Problem 2: Artefakte sind auf unterschiedlichen Abstraktionsebenen und daher nicht integrierbar

Die erstellten Artefakte befinden sich auf unterschiedlichen Abstraktionsebenen, beschreiben unterschiedliche Ausschnitte und sind deswegen untereinander nicht einfach integrierbar.

Die Integration in die Anforderungsspezifikation bzw. dem Lastenheft findet häufig nur textuell statt.

5.3.1 Entstehung bzw. Ursache des Problems

Artefakte werden von unterschiedlichen Mitarbeitern unterschiedlich erstellt. Gerade in der Modellbildung zeigt sich, dass die entstehenden Modelle äußerst unterschiedlich sind, auch wenn sie den gleichen oder einen mit einem anderen Modell überschneidenden Teilbereich beschreiben.

Zusätzlich gibt es in der Praxis keine Vorgaben zur Auswahl des relevanten Wissens, das zu modellieren ist und ebensowenig Aussagen bezüglich einer angestrebten Abstraktionsebene. Fehlendes Wissen, das es ermöglichen würde, die Modelle zu integrieren, wird nicht erfragt. Dies führt dazu, dass erfasste Inhalte in Artefakten häufig mehrdeutig sind.

5.3.2 Beispiel

Abbildung 5.6 zeigt ein triviales Beispiel, betreffend die Schwierigkeit, wenn zwei Modelle integriert werden sollen, aber jeweils einen anderen Ausschnitt enthalten. Die linke Seite zeigt etwas das *Disposition* heißt und *Disponent* aggregiert. Auf der rechten Seite existiert eine *Debitorenbuchhaltung* und *Buchhalter*.



Abbildung 5.6: Beispiel: Zwei unterschiedliche Modelle, die jeweils eine Facette eines Unternehmens beschreiben

Es gibt nun verschiedene Möglichkeiten diese Modelle zu integrieren. Grundsätzlich überschneiden sie sich offensichtlich nicht, außer der Disponent kann auch ein Buchhalter sein und umgekehrt. Dies bedeutet zur Integration kann dies z. B. über eine übergeordnete organisatorische Einheit geschehen, oder über eine Beziehung, zwischen Disposition und Debitorenbuchhaltung. Z. B. könnte eine Beziehung existieren, die aussagt, dass die Debitorenbuchhaltung der Disposition die Zustellung einer Ware nach Zahlungseingang erlaubt. Hier ist noch vieles mehr vorstellbar.

Es zeigt, dass die Integration der Modelle äußerst schwierig ist, selbst bei so einem trivialen Beispiel. Hinzu kann die Schwierigkeit kommen, dass sich die Entitäten inhaltlich überschneiden. Die offenen Fragen müssen bei Integration geklärt werden. Häufig sind die Probleme umfangreich und eine Integration zieht viel Arbeitsaufwand nach sich.

Wie in unserer Studie erhoben, werden die Modelle typischerweise unabhängig voneinander erstellt. Dies bedeutet, es gibt keine Vorgaben welche Modelle mit welchen Inhalten erstellt werden. Wenn nun noch hinzu kommt, dass Abläufe beschrieben werden, ist dies der Struktur, wie im Beispiel, noch schwieriger zuordenbar.

Abbildung 5.7 zeigt ein Sequenzdiagramm, in dem die Interaktion zwischen Benutzer und System dargestellt wird, während auf der rechten Seite ein Geschäftsprozess die zu verrichteten Aktivitäten unabhängig vom System und auf einer anderen Abstraktionsebene darstellt. Der Zusammenhang zwischen dem Sequenzdiagramm und dem dargestellten Geschäftsprozess ist nicht mehr einfach eruiert. Das hinter *Auftragsannahme prüfen* der Geschäftsprozess steckt, der im Sequenzdiagramm auf der linken Seite dargestellt ist, ist nicht einfach erkennbar. Dies liegt u.a. daran, dass verschiedene Rollen angegeben sind, da die Kreditwürdigkeit z. B. vom Buchhalter geprüft wird, der Auftrag aber initial die Disposition erreicht. Hinter dem Geschäftsprozess *Auftragsannahme prüfen* kann daher durchaus eine andere Organisationseinheit verantwortlich sein. Die Modelle sind dennoch nicht einfach integrierbar.

5.3.3 Auswirkungen

Die Auswirkungen dieses Problems sind vielfältig. Inkonsistenzen zwischen den Artefakten können z. B. aufgrund der unterschiedlichen Abstraktionsebenen der Inhalte der Artefakte nicht gefunden werden, da die Artefakte nicht integriert werden können, ohne die Lücke dazwischen zu schließen.

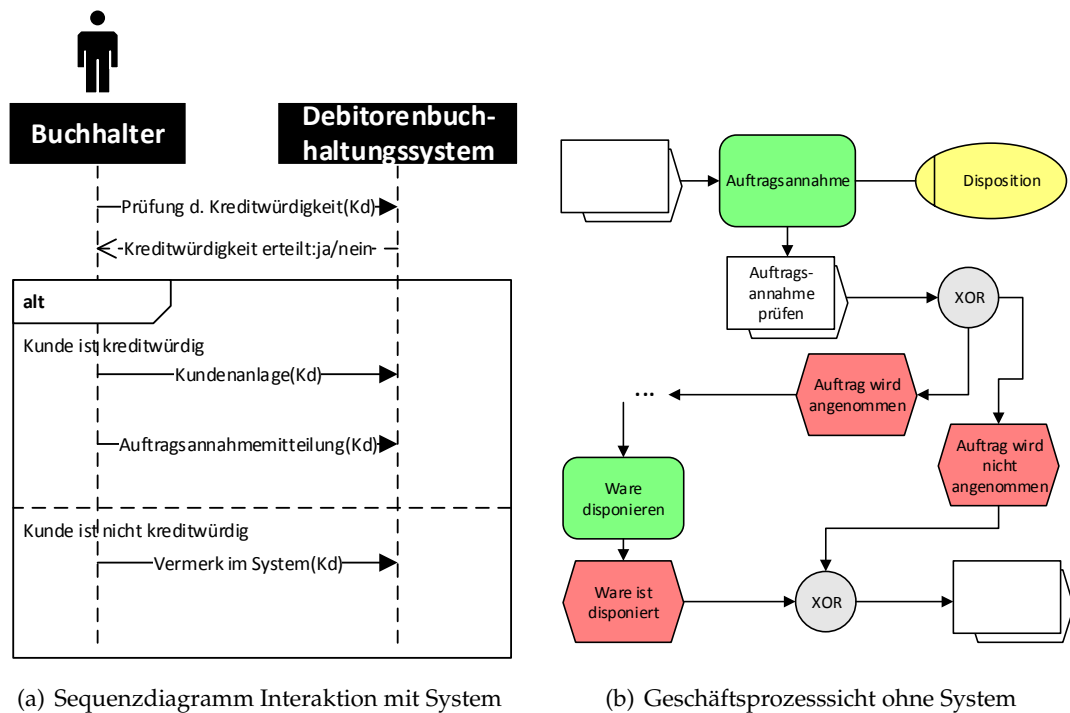


Abbildung 5.7: Sequenzdiagramm und Geschäftsprozess zeigen gleichen Vorgang aus unterschiedlichen Perspektiven und unterschiedlichen Abstraktionsebenen

Abstrakte Inhalte, die nicht zuzuordnen sind, führen zu missverständlichen Interpretationen, die sich auf das zu implementierende Softwaresystem auswirken und ggf. erst sehr spät im Softwareentwicklungsprozess gefunden werden können.

5.3.4 Mögliche Lösung des Problems

Ein Lösungszustand sieht so aus, dass alle in frühen Phasen erstellten Artefakte inhaltlich in eine Beziehung zueinander gesetzt werden können. Und somit das Wissen das in ihnen expliziert wurde, in der Anforderungsspezifikation bzw. im Pflichtenheft integrierbar ist und Ungereimtheiten, z. B. Widersprüche, einfacher identifizierbar sind, als ohne diese Beziehung unter den Modellen.

Lösungsidee. Eine mögliche Lösung, ohne dabei vorhandene Modellierungstechniken zu ersetzen, besteht darin, ein Modell, das diese verschiedenen Sichten und Abstraktionsebenen in den Modellen integriert, als Basis zur Erstellung von weiteren Modellen in dieser Phase zu verwenden. Die Entitäten des weiteren Modells sind immer auf die des initialen Modells zurückführbar. Dieses Modell nennen wir Basismodell.

Das Basismodell enthält Entitäten auf unterschiedlichen Abstraktionsebenen und die Verknüpfung zu Inhalten anderer Artefakt-Typen ist spezifiziert. Wird die Verknüpfung zu anderen Artefakt-Typen definiert, sodass zumindest die Entitäten übernommen werden können, ist ein inhaltlicher Zusammenhang identifizierbar.

Werden zusätzlich Aussagen darüber getroffen, wann ein angestrebtes Abstraktionsniveau erreicht ist, das die Verknüpfung zu anderen Artefakten erlaubt, werden alle vom initialen Modell (dem Basismodell) abhängigen Artefakte mit den gleichen Entitäten starten. Daher finden sich die Entitäten eines Basismodells in allen Artefakten wieder.

Beispiel zur Lösungsidee. Wir nehmen an, dass sich beim Erstellen des Basismodells herausgestellt hat, wie die Struktur der Prozesse ist, welche Akteure involviert sind und welche Teilsysteme Unterstützung bei der Durchführung der Geschäftsprozesse leisten sollen.

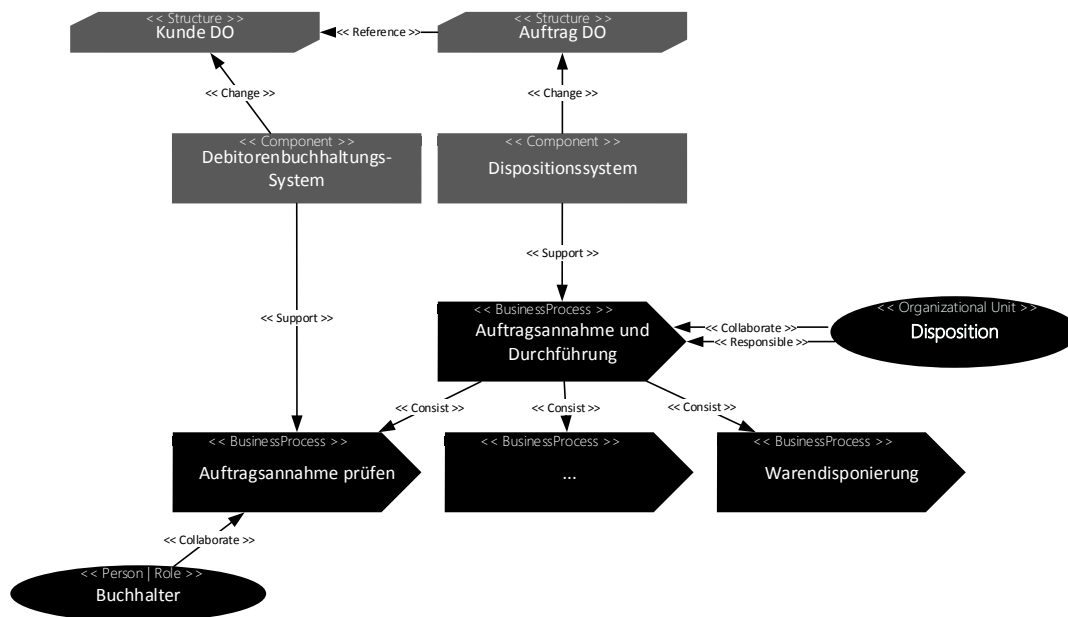


Abbildung 5.8: Zeigt ein Basismodell für die Integration der Modelle aus dem Beispiel aus Abbildung 5.7

Abbildung 5.7 zeigt ein Beispiel, bei dem zwei unterschiedliche Modelle Ausschnitte einer Domäne beschreiben, die zwar miteinander zu tun haben, aber aus den Modellen nicht klar erkennbar sind. Abbildung 5.8 zeigt ein Basismodell, das als Basis für die Erstellung dieser Modelle verwendet werden kann. Hierbei würde das in Abbildung 5.7 in Teil (a) dargestellte Sequenzdiagramm den Geschäftsprozess *Auftragsannahme prüfen* darstellen und die Interaktion zwischen Teilsystem *Debitorenbuchhaltungssystem* und dem Akteur *Buchhalter* präzisieren.

In diesem Beispiel ist auch klar erkennbar, wie das Geschäftsprozessmodell in Abbildung 5.7 in Teil (b) in das Gesamtbild integrierbar ist. Die organisatorische Einheit *Disposition* führt den Geschäftsprozess *Auftragsannahme und Durchführung* aus und verantwortet diesen. Durch dieses Basismodell ist (i) die Rolle zwischen System und Akteuren klar und (ii) die Rolle der Modelle klar, die unterschiedliche Ausschnitte beschreiben. So beschreibt das Modell in Teil (b) in Abbildung 5.7 die Interaktion zwischen Akteur und System und der Teil (b) die Aktivitäten in einem Geschäftsprozess. Diese Modelle sind durch das Basismodell in Abbildung 5.8 einfach integrierbar, da deren Zusammenhang klar wird.

5.3.5 Kriterium

Um die Abstraktionsebenen von Artefakten zu überprüfen, beziehen wir die Artefakte zueinander. Wir legen fest, dass Artefakte dann auf einer Abstraktionsebene sind, wenn diese so in Beziehung setzbar sind, dass keine Lücken vorhanden sind, die durch Interpretation geschlossen werden müssen.

Daraus leiten wir folgendes Kriterium ab:

- ▶ **K2.1:** Die Artefakte stehen in einer Beziehung zueinander, sodass keine Interpretationslücken geschlossen werden müssen.

5.4 Problem 3: Inkonsistente Artefakte bzw. die Konsistenz der Artefakte kann nicht überprüft werden

Artefakte, erstellt in frühen Phasen des Requirements Engineering, können typischerweise nicht auf Konsistenz untereinander geprüft werden. Ebenso wenig können aufgrund fehlender Formalismen auch einzelne Artefakte nicht auf Konsistenz überprüft werden.

5.4.1 Entstehung bzw. Ursache des Problems

Artefakte in frühen Phasen des Requirements Engineerings werden ohne Vorgaben erstellt. Diese erstellten Artefakte sind typischerweise semi-formale Modelle (z. B. auf Basis von UML-Klassendiagrammen) oder auf Basis von Textschablonen (z. B. Use-Cases/Szenarien) erstellt wurden.

Dies liegt daran, dass in diesen initialen Artefakten noch keine Formalismen eingesetzt werden, da das Erarbeiten einer gemeinsamen Vorstellung von Problem und Lösung im Mittelpunkt steht. Dies gilt aber nicht, wenn die initiale Phase in die Spezifikationsphase übergeht und Detaillierung stattfindet.

Ob Artefakte inkonsistent sind, stellt sich nach Aussagen (siehe Kapitel 3) häufig erst später im Entwicklungsprozess heraus, da die Konsistenzsicherung nicht im Rahmen des Requirements Engineerings vollständig durchführbar ist. Dies liegt daran, dass mehrere Personen an der Erstellung der Anforderungsspezifikation beteiligt sind und Widersprüche nicht innerhalb eines Abschnittes zu finden sein müssen, die von einer einzelnen Person überprüft bzw. bearbeitet wurde.

5.4.2 Auswirkungen

Artefakte, erstellt in diesen frühen Phasen werden häufig als Grundlage zur Spezifikation verwendet (z. B. werden auf Basis dieser frühen Artefakte Aktivitätsdiagramme erstellt). Dies geschieht aber nicht nur einmal, sondern häufiger im Requirements Engineering. Da alle initial erstellten Artefakte nicht überprüft werden können, können modellierte Widersprüche teilweise erst bei Integration der Artefakte in die Anforderungsspezifikation bzw. dem Pflichtenheft oder erst später in der Design- und/oder Implementierungsphase gefunden werden.

5.4.3 Mögliche Lösung des Problems

Eine mögliche Lösung ist die Verwendung eines Formalismus, der es erlaubt, die erstellten Modelle auf Konsistenz zu prüfen. Es entstehen in frühen Phasen viele Modelle und viele entstehen ohne die Möglichkeit die Modellinhalte formal zu definieren.

Lösungsidee. Ein Modell (das Basismodell), dient als Basis für die Wahl von Entitäten in anderen Artefakten und wird konsistent gehalten. Konsistent bedeutet in diesem Zusammenhang, dass es keine Widersprüche zwischen den Aussagen die im Basismodell erfasst wurden und den Inhalten der darauf aufbauenden Artefakte gibt. Ergeben sich Änderungen durch darauf aufbauende Artefakte im Basismodell, muss das Basismodell konsistent bleiben. Von den Änderungen betroffene Artefakte müssen gegebenenfalls angepasst werden.

Beispiel zur Lösungsidee. Abbildung 5.9 zeigt ein einfaches Beispiel. Der *Einkaufsvorgang* gibt an, dass die *Artikelliste* nur lesend (*Read-only-use*) verwendet wird. Diese Verwendung kann z. B. in einem Geschäftsprozess detailliert sein. Ein Teilprozess *Übernahme* des Prozesses *Einkaufsvorgang* gibt an, dass die *Artikelliste* geändert wird.

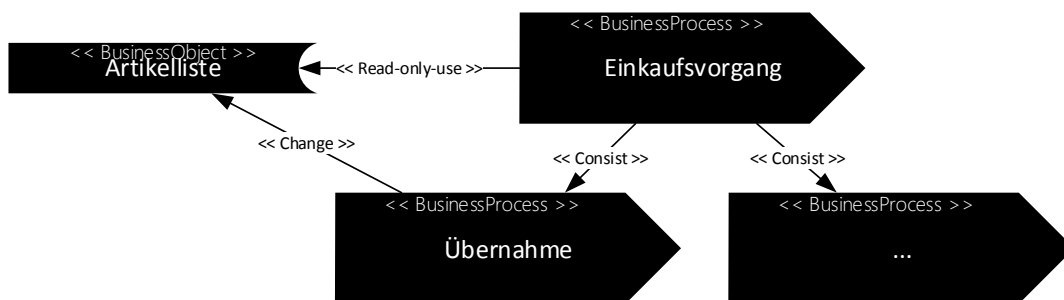


Abbildung 5.9: Mögliches Szenario, um durch Ableitung auf Inkonsistenzen zu schließen

Im Rahmen der Ableitungsregeln wird im Basismodell darauf geschlossen, dass wenn *Übernahme* die *Artikelliste* ändert, dies auch der übergeordnete Prozess *Einkaufsvorgang* tut. Alle Artefakte, die die Beziehung zwischen *Einkaufsvorgang* und *Artikelliste* detaillieren, müssen daraufhin überprüft werden. Die Änderung der Involviertheit von *Artikelliste* kann sich nun auf die abgebildeten Strukturen auf Softwareseite, sowie der Unterstützung der Software des *Einkaufsvorganges* auswirken. Dies muss überprüft werden, um die Konsistenz der Artefakte sicher zu stellen.

Bereits dieses einfache Beispiel zeigt, dass sich kleine Änderungen im Basismodell auf die Detaillierung vieler davon abhängiger Artefakte auswirken können. Im Requirements Engineering bedeutet dies: Wird ein Artefakt, das z. B. Ausschnitte des oben genannten Szenarios beschreibt, geändert und scheinen diese Änderungen nur geringer zu sein, können sich tiefgreifende Auswirkungen auf die zu entwickelnde Softwarelösung ergeben.

Dieses Sicherstellen der Konsistenz von Modellen in frühen Phasen des Requirements-Engineering-Prozesses ist besonders wichtig, da die Fehler, die in diesen Mo-

dellen gemacht werden, teilweise erst sehr spät gefunden werden (z. B. erst bei Integration in die Anwendungsdomäne).

5.4.4 Kriterium

Wir sagen, dass die Artefakte, erstellt in frühen Phasen des Requirements Engineerings, dann konsistent sind, wenn sie in keinem Widerspruch stehen.

Wir leiten daraus folgendes Kriterium ab:

- ▶ **K3.1:** Die Artefakte, erstellt in frühen Phasen des Requirements Engineerings, stehen in keinem Widerspruch untereinander.

5.5 Problem 4: Unterschiedliche Modellbildung durch Modellierer

5.5.1 Entstehung bzw. Ursache des Problems

Modelle werden von Modellierern unterschiedlich erstellt. In der Modellbildung spielen subjektive Einflüsse eine Rolle. In der Praxis gibt es keine Vorgaben zur Auswahl des zu modellierenden Wissens. Zusätzlich werden die Abstraktionsebenen subjektiv gewählt.

Dies führt zu unterschiedlichen Modellen, auch wenn der gleiche oder ein sich überschneidender Sachverhalt dargestellt wird.

5.5.2 Beispiel

In der durchgeführten Analyse (siehe Abschnitt 4) der Inhalte von Artefakten in frühen Phasen, enthielten 73% der analysierten Modelle Informationen, die einem Geschäftsprozess entsprachen.

Dies ist dann der Fall, wenn (a) eine benannte Beziehung eine Involviertheit in einem Vorgang andeutet (z. B. *erstellt*), (b) eine Entität direkt einen Geschäftsprozess benennt (z. B. Rechnungsstellung) oder (c) mit Funktions-Stub eine Verarbeitung angedeutet wird (z. B. *VerarbeiteRechnung*). Siehe dazu Abbildung 5.10.

Alle drei der genannten Darstellungsformen von Geschäftsprozessen kamen in den Artefakten vor. Unsere durchgeführte Studie zeigt (siehe dazu Kapitel 4), dass in 50% der Fälle Fall (a), in ca. 29% Fall (b) und in ca. 21% der Fall (c) in der Praxis angetroffen wird.

5.5.3 Auswirkungen

Aus dem geschilderten Sachverhalt ergeben sich mehrere Probleme:

- (i) Je nach gewählter Repräsentation eines Vorganges zur Erarbeitung eines Problemverständnisses, werden Entitäten überhaupt nicht genannt, z. B. wird in

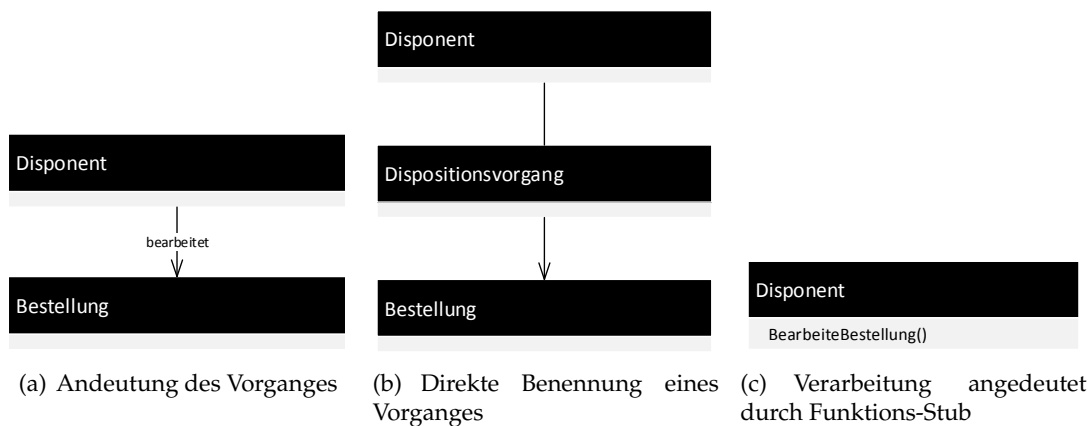


Abbildung 5.10: Beispiele für unterschiedliche Modellbildung des gleichen Sachverhaltes

Fall (c) in der Abbildung 5.10 die Entität Bestellung nicht direkt modelliert. Durch die Namensgebung der Funktion ist dies indirekt geschehen, aber nicht explizit.

- (ii) Die Integration der Vorgänge z. B. zu einer Prozesslandschaft ist nicht einfach möglich, da die Prozesse unterschiedlich dargestellt werden und teilweise die Identifikation dadurch erschwert wird.
- (iii) Modelle sind nicht gut zur Kommunikation geeignet, da gleiche Sachverhalte unterschiedlich modelliert wurden und dies je nach Leser des Modells durch den verwendeten Syntax durchaus auch andere Bedeutungen haben kann.

5.5.4 Mögliche Lösung des Problems

Alle Artefakte werden von den unterschiedlichen Modellierern *ähnlich* erstellt.

Dieses generische Ziel ist für alle Artefakte, die im Laufe des Requirements Engineerings erstellt werden nicht einfach zu erreichen. Was erreichbar ist, ist den Modellbildungsvorgang zu leiten, indem die Typen, die Entitäten bilden sollen, explizit genannt werden.

Lösungsidee. Die Entitäten im Basismodell dienen zur Benennung der Entitäten in anderen Artefakten. Sie stellen einen *Startpunkt* dar, mit dem begonnen wird, die Beziehungen und Entitäten im Basismodell mit anderen Modellierungstechniken weiter zu detaillieren.

Beispiel zur Lösungsidee. Dieses Beispiel zeigt die Leitung der Modellbildung zweier Sequenzdiagramme auf Basis des Basismodells.

Das Beispiel in Abbildung 5.11 zeigt die Vorgaben, dass zu Detaillierung der Beziehung zwischen Komponenten und Geschäftsprozess immer drei Schritte notwendig sind (a) die Auswahl mindestens eines Geschäftsprozesses, (b) die Identifikation der Akteure, die die in (a) ausgewählten Geschäftsprozesse ausführen und (c) die Identi-

fikation der Komponenten, die die in (a) ausgewählten Geschäftsprozesse unterstützen.

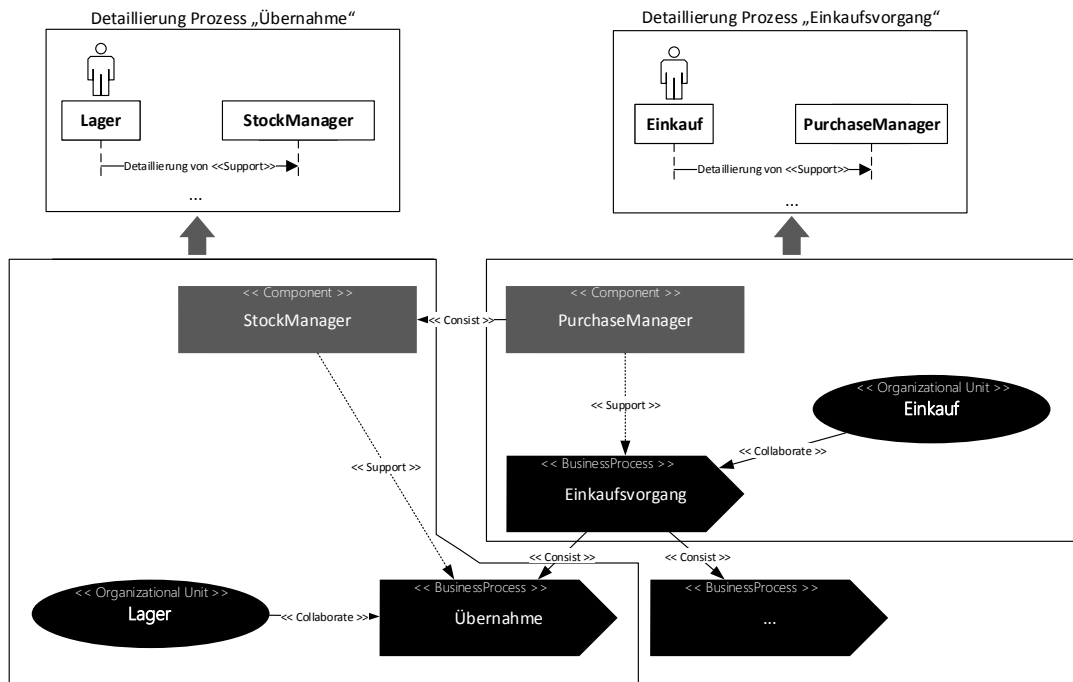


Abbildung 5.11: Leitung der Modellbildung

Die Akteure aus (b) sowie die Komponenten aus (c) werden dabei zu den Objekten im Sequenzdiagramm, die die Geschäftsprozesse aus (a) detaillieren. Die Detaillierung muss immer konform zu den dargestellten Beziehungen im Basismodell sein. Dies führt dazu, dass (i) die Auswahl von Akteuren geleitet wird und (ii) der abstrakte Inhalt der Interaktion zwischen den Akteuren durch das Basismodell, zumindest grob, gegeben ist.

Die Beziehungen zwischen den Entitäten im Basismodell dienen dabei als Basis für weitere Modellierungstechniken. So können z. B. die Beziehungen zwischen Komponenten mit Hilfe von Klassen- und/oder Komponentendiagrammen detailliert werden. Die Komponenten des Basismodells dienen dann als Entitäten für die Klassen- bzw. Komponentendiagramme. Die Beziehungen zwischen den Komponenten müssen sich in den Klassen- bzw. Komponentendiagrammen in ihrer Bedeutung wieder finden.

5.5.5 Kriterium

Wie in diesem Abschnitt erläutert, werden Artefakte in frühen Phasen von verschiedenen Personen unterschiedlich erstellt, wofür es viele Ursachen gibt (z. B. keine Vorgaben zur angestrebten Abstraktionsebene). Wir betrachten das Problem dann als gelöst, wenn von verschiedenen Personen Artefakte ähnlich erstellt werden.

Daraus leiten wir folgendes Kriterium ab:

- **K4.1:** Die Artefakte, die gleiche Ausschnitte eines Problems bzw. einer Lösung

mit Hilfe der gleichen Modellierungstechnik beschreiben, werden von unterschiedlichen Personen ähnlich modelliert.

Eine Ähnlichkeit ist dann gegeben, wenn bestimmte Merkmale übereinstimmen [Dud11]. Wir legen fest, dass Modelle dann ähnlich sind, wenn ihre Entitäten (Klassen, Akteure, Komponenten, ...) und die Bedeutung ihrer Beziehungen übereinstimmt.

5.6 Problem 5: Wissen ist unausgeglichen repräsentiert

Modellierer erfassen in Modellen Wissen aus unterschiedlichen Gründen und mit unterschiedlicher Priorität. So werden häufig leicht erfassbare Sachverhalte in einem höheren Detaillierungsgrad beschrieben, wie es nicht so einfach zugängliche Sachverhalte werden. Einzelne Sachverhalte sind daher in Artefakten häufig unter- oder überrepräsentiert.

5.6.1 Entstehung bzw. Ursache des Problems

Beim Erfassen von Artefakten durch den Erheber mit Hilfe von Domänenexperten wird das Wissen erfasst, das der Domänenexperte als wichtig betrachtet. Häufig wird integratives Wissen, das zwei Ausschnitte eines Problembereiches verbinden würde, nicht erfasst, da dies außerhalb der Expertise des Domänenexperten liegt und dem Erheber das Vorhandensein des integrativen Wissens auch unbewusst bzw. unbekannt sein kann.

5.6.2 Beispiel

Die Abbildung 5.12 zeigt uns jeweils einen kleinen Teil des Wissens über die Struktur einer Disposition sowie des Einkaufs.

Beispielszenario: Wie in Abbildung 5.12 ersichtlich ist im Dispositionsvorgang eine Bestellung involviert. Wird die Ware ausgeliefert und es sind nur mehr wenige Exemplare eines Artikels vorhanden, soll der Bestellvorgang des Einkaufes angestoßen werden.

Nun kann viel über den Bestellvorgang des Einkaufes bekannt sein (z. B. wie werden Bestellungen ausgeschrieben, wie werden Rechnungen übermittelt, wie werden die Verbindlichkeiten verbucht, usw.), der Zusammenhang zwischen diesen scheinbar unabhängigen Problembereichen bleibt verborgen, wenn diese Bereiche nur isoliert betrachtet werden, was in der Praxis häufig geschieht.

5.6.3 Auswirkungen

Dieses nicht erfasste bzw. nur indirekt vorhandene Wissen fehlt häufig. Da genug Informationen vorhanden sind, um Anforderungen der detaillierten Ausschnitte zu implementieren und es keine Möglichkeit gibt, auf dieses fehlende Wissen zu schließen, ist das Fehlen den Entwicklern auch nicht bewusst.

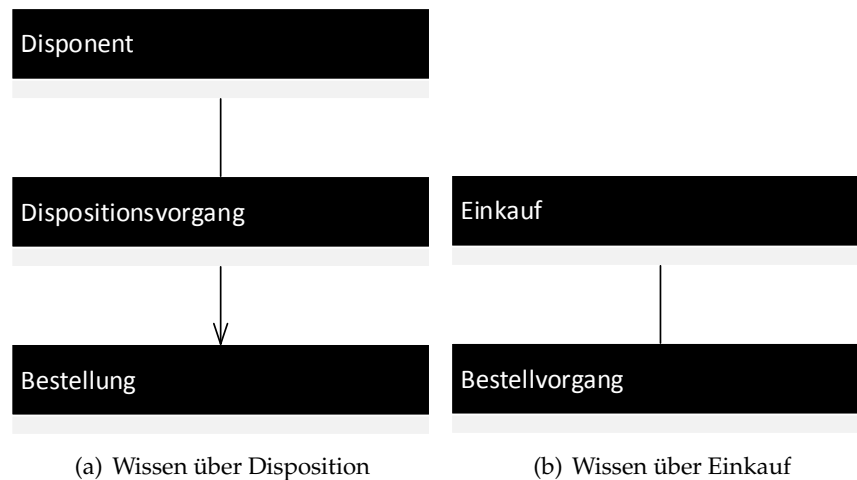


Abbildung 5.12: Zwei scheinbar unabhängige Bereiche einer beispielhaften Problemdomäne

Eines der Probleme, das auch in Abschnitt 5.8 angesprochen wird, ist die fehlende Vorstellung über das Gesamtsystem von Entwicklern. Dieses fehlende Wissen ist notwendig, um diese Ausschnitte zu identifizieren.

Das Fehlen dieser Ausschnitte hat weitreichende Auswirkungen. Da diese Fehler erst sehr spät im Entwicklungsprozess gefunden werden, führen sie zu erheblichen Mehrkosten.

5.6.4 Mögliche Lösung des Problems

Ein zusammenhängendes Modell der Problem- und Lösungsdomäne expliziert die Zusammenhänge zwischen Lösungs- und Problemdomäne für die spätere Detaillierung.

Das zusammenhängende Modell (das Basismodell) bietet den Domänenexperten zusätzlichen Hinweis auf eventuell noch nicht erfasstes relevantes Wissen.

Lösungsidee. Ein Basismodell, das Wissen über das Gesamtsystem mit Umgebung in Form der Geschäftsprozesse enthält, dient zur Identifikation von noch nicht behandelten (oder noch nicht genug detaillierten) Ausschnitten. Primäres Ziel ist das Schaffen einer Gesamtsystemvorstellung in Form des Basismodells auf einer hohen Abstraktionsebene um die Übersicht zu wahren.

Beispiel zur Lösungsidee. Das Beispiel zeigt den Identifikationsvorgang von nicht näher behandelten Beziehungen im Basismodell.

Abbildung 5.13 zeigt ein an das Beispiel in Abbildung 5.12 angelehntes Basismodell. Verschiedene Ausschnitte des Basismodells werden mit unterschiedlichen Modellierungstechniken detailliert. So wird in der Abbildung 5.13 der Geschäftsprozess *Bestellvorgang* in einem Sequenzdiagramm detailliert. Hierbei gilt es auch involvierte Strukturen zu beachten (in diesem Fall *Bestellung* DO). Der Geschäftsprozess *Dispo-*

sitionsvorgang und die davon angestoßenen Geschäftsprozesse werden z. B. mit Hilfe einer ereignisgesteuerten Prozesskette weiter detailliert.

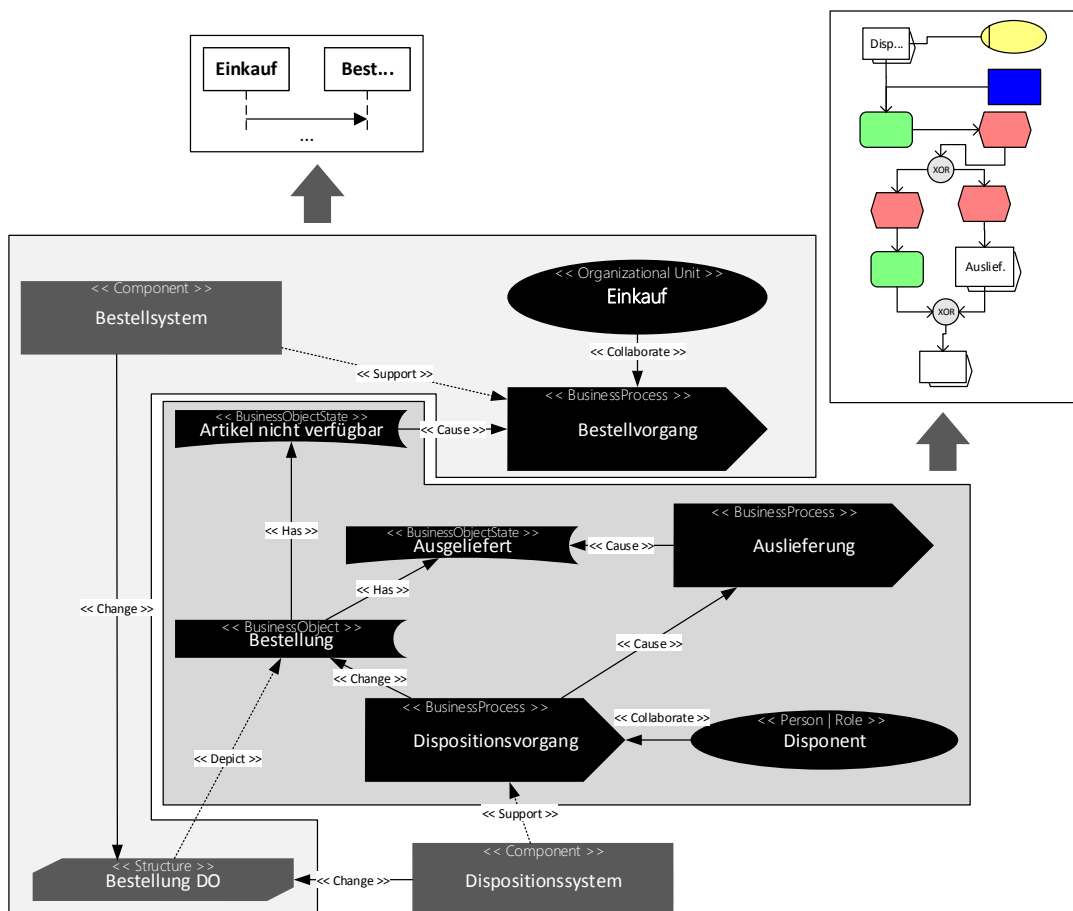


Abbildung 5.13: Identifikation von nicht behandelten Beziehungen

Sind die Modelle erstellt, lassen sich mit Hilfe des Basismodells relativ einfach die nicht behandelten Beziehungen identifizieren, z. B. die Unterstützung des *Dispositionsvorganges* durch das *Dispositionssystem*.

In dem Beispiel wird das Auslösen des *Bestellvorganges* durch das Erreichen des Zustandes *Artikel nicht verfügbar* nicht behandelt. Hier fehlt Domänenwissen, das noch erfasst werden muss.

Mit Hilfe des Modellierungsansatzes versuchen wir jene Entitäten auf einer Abstraktionsebene zu erfassen, die dann später detailliert werden können, wie es in diesem Beispiel auch gezeigt wird. Das entstehende abstrakte Begriffsnetz liefert Überblickinformationen, in Form von Geschäftsprozessen, über das System und der Umgebung und gibt ebenso Hinweise auf noch fehlende Informationen.

5.6.5 Kriterium

Wir nehmen an, es gibt in einem Problembereich, der von einer Software unterstützt werden soll, einen Gesamtzusammenhang. Der Gesamtzusammenhang beschreibt den Problem- und den Lösungsbereich.

Die in frühen Phasen des Requirements Engineering erhobenen Artefakte beschreiben verschiedene Teile dieses (teilweise nur mental vorhandenen) Gesamtzusammenhanges. Wir sagen, dass das Wissen einer Erhebung dann ausgeglichen ist, wenn alle Ausschnitte des Gesamtzusammenhanges von zumindest einem Artefakt beschrieben werden.

Ein Ausschnitt in diesem Kontext, stellt zumindest einen Zusammenhang dar. Der Zusammenhang stellt eine Beziehung zwischen zumindest zwei u.a. Sachverhalten her [Dud11].

Daraus ergibt sich folgendes Kriterium:

- ▶ **K5.1:** Alle Ausschnitte eines Gesamtzusammenhanges werden von zumindest einem Artefakt beschrieben.

5.7 Problem 6: Keine Verknüpfung von Artefakten zu anderen Artefakten

5.7.1 Entstehung bzw. Ursache des Problems

Wie aus den Ergebnissen der durchgeführten Studie in Abschnitt 3 ersichtlich ist, werden Artefakte in frühen Phasen unabhängig voneinander erstellt. Dies führt u.a. dazu, dass sich überlappende und/oder gleiche Sachverhalte mit unterschiedlichen Notationen repräsentiert werden.

Die in frühen Phasen erstellten Artefakte dienen als Quelle für die Anforderungsspezifikation und/oder werden in diese integriert. Weiters dienen Artefakte dem initialen Schaffen eines Problem- und Lösungsverständnisses und werden in die Anforderungsspezifikation bzw. dem Pflichtenheft integriert.

5.7.2 Beispiel

Abbildung 5.14 zeigt ein Sequenzdiagramm, in dem die Interaktion zwischen Benutzer und System dargestellt wird, während auf der rechten Seite ein Geschäftsprozess die zu verrichteten Aktivitäten unabhängig vom System darstellt. Der Zusammenhang zwischen Geschäftsprozess und Sequenzdiagramm ergibt sich dann, wenn es möglich ist, den involvierten Teil des Systems und des Benutzers mit dessen Aufgabe im Geschäft in Beziehung setzen zu können.

In diesem Beispiel ist der Zusammenhang noch aufgrund der Verwendung der gleichen Bezeichner für Vorgänge ersichtlich. Dabei betrifft der Schwerpunkt in den Geschäftsprozessen die Darstellung der Aktivitäten und nicht die Interaktion mit dem System. Dies wird erst dann ein Problem, wenn viele Modelle vorhanden sind und die Zusammengehörigkeit der Modelle nicht mehr einfach feststellbar ist.

5.7.3 Auswirkungen

Das Fehlen einer Verknüpfung führt zu einer Vielzahl von Problemen:

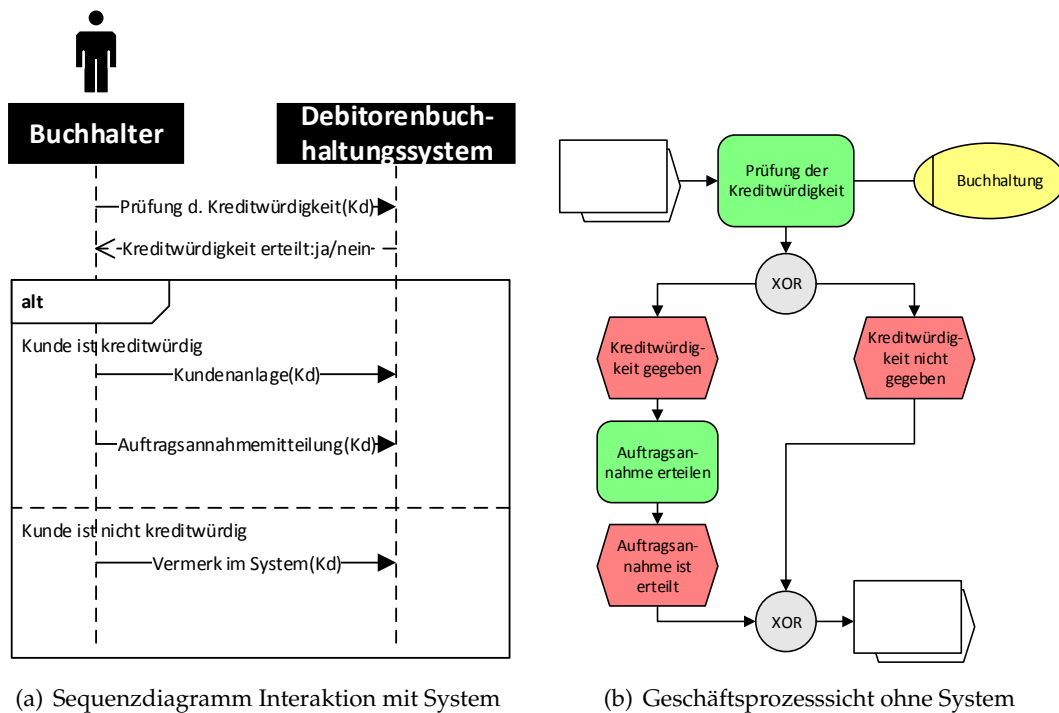


Abbildung 5.14: Sequenzdiagramm und Geschäftsprozess zeigen gleichen Vorgang aus unterschiedlichen Perspektiven

Die in frühen Phasen des Requirements Engineering erstellten Artefakte, die losgelöst voneinander erstellt werden, können nicht in Beziehung zueinander gesetzt werden. Dies ist insbesondere dann ein Problem, wenn sich Inhalte in Artefakten ändern, die auch Auswirkungen auf damit in Zusammenhang stehende Artefakte haben.

Für Stakeholder ist es schwierig, sich in den Problem- und Lösungsbereich hinein zu arbeiten, da der Zusammenhang zwischen den Artefakten nicht ersichtlich ist.

Die Artefakte, die in die Anforderungsspezifikation bzw. das Pflichtenheft integriert werden, müssen inhaltlich überprüft werden. Dies wird häufig aufgrund der Anzahl von Artefakten nicht zusätzlich gemacht – überprüft wird nur die inhaltliche Konsistenz von Einzelartefakten.

5.7.4 Mögliche Lösung des Problems

Es gibt eine definierte Verknüpfung zu anderen Artefakten. Für das Beispiel aus Abbildung 5.14 würde dies bedeuten, dass der Buchhalter der in Interaktion mit dem System steht, aus (a) genau die Schritte mit dem System durchführt, die im allgemeineren Geschäftsprozess aus (b) ebenso expliziert wurden.

Eine Möglichkeit dies zu bewerkstelligen ist, die drei Entitäten, die zur Identifizierung des Geschäftsprozesses, des (Teil-)Systems und des Akteurs nötig sind explizit zu benennen. Im Beispiel aus Abbildung 5.14 würde dies bedeuten, es gibt ein (Teil-)System, das den Geschäftsprozess, den wir hier *Kreditwürdigkeitsprüfung für Auftragsannahme* nennen, unterstützt. Ein Akteur ist für die Ausführung des Ge-

schäftsprozesses zuständig. Diese Konstellation ermöglicht die Identifikation des Zusammenhangs der beiden Modelle, da gesagt werden kann, das Sequenzdiagramm in der Abbildung 5.14 in Teil (a) ist eine Detaillierung der Unterstützung des Systems im Geschäftsprozess, durchgeführt durch den Akteur *Buchhalter*. Für Teil (b) ist es eine Detaillierung der Ausführung des Geschäftsprozesses *Kreditwürdigkeitsprüfung für Auftragsannahme* durch den zuständigen Bearbeiter *Buchhalter*. So kann ein direkter Zusammenhang zwischen den Modellen, um eine Hilfskonstruktion, die wir in dieser Arbeit Basismodell nennen, hergestellt werden. Dies erlaubt uns (a) die Modelle in eine Beziehung zu setzen und (b) die betrachteten Entitäten in einem Modell vor der Erstellung einzuschränken.

Lösungsidee. Für das Basismodell werden Vorgaben zur Verknüpfung zu gängigen Artefakt-Typen definiert, die in frühen Phasen des Requirements Engineerings eingesetzt werden. Dies betrifft vor allem die Vertreter der OOAD sowie von Geschäftsprozessmodellierungstechniken und Organisationsmodellen.

Beispiel zur Lösungsidee. Das Beispiel zeigt zwei ausgewählte Bereiche im Basismodell die für eine definierte Verknüpfung notwendig sind.

Abbildung 5.15 zeigt einerseits den Bereich bestehend aus *Bestellsystem*, *Bestellung DO*, *Einkauf* und *Bestellvorgang*, den wir Bereich 1 nennen und den Bereich bestehend aus *Einkauf*, *Bestellvorgang* und *Bestellung*, den wir Bereich 2 nennen.

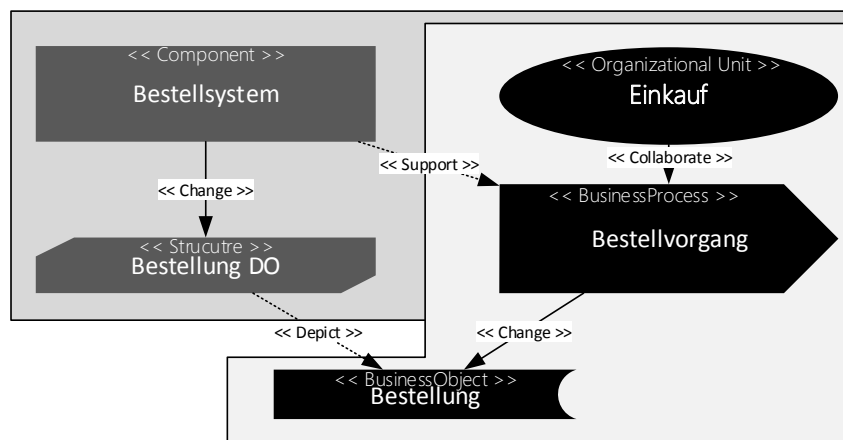


Abbildung 5.15: Beispiel Auswahl von Entitäten im Basismodell für eine Verknüpfung

Bereich 1 ist eine mögliche Auswahl für die Verknüpfung zu Sequenzdiagrammen. *Einkauf* als organisatorische Einheit ist ein Akteur, der mit dem System *Bestellsystem* im Rahmen des Geschäftsprozesses *Bestellvorgang* interagiert. Eine Vorgabe zur Auswahl des *Bestellvorganges* könnte sein, dass der Geschäftsprozess so weit detailliert (also aus anderen Geschäftsprozessen zusammengesetzt) sein muss, bis ein Geschäftsprozess nur mehr aus der Interaktion mit dem System besteht. Dann ist die Auswahl des Geschäftsprozesses *Bestellvorgang* für eine Verknüpfung zum Sequenzdiagramm detailliert genug. Im Sequenzdiagramm, muss sich die Art der Änderung (*Change*) den involvierten *Bestellung DO* im Laufe der Interaktion mit dem System ergeben.

Bereich 2 eignet sich zur Detaillierung in Geschäftsprozessmodellierungstechniken. Auch hier muss die Änderung der *Bestellung* mit modelliert werden.

5.7.5 Kriterium

In frühen Phasen des Requirements Engineerings werden verschiedene Artefakt-Typen zur Repräsentation des erhobenen Wissens verwendet. Die erstellten Artefakte stehen häufig nicht in einer Beziehung. Wir leiten daraus ab, dass dieses Problem dann gelöst ist, wenn das folgende Kriterium erfüllt ist:

- **K6.1:** Eine Verknüpfung zu anderen Artefakten ist angemessen möglich.

5.8 Problem 7: Kein stakeholderweites Verständnis

In Softwareprojekten herrscht kein stakeholderweites Verständnis über die Problem- und Lösungsdomäne. Dies hat weitreichende Auswirkungen. Dieser Abschnitt beschreibt die von uns gefundenen Gründe und Auswirkungen.

5.8.1 Entstehung bzw. Ursache des Problems

In der von uns durchgeführten Studie haben wir Unternehmen befragt, die Individualsoftware entwickeln und häufig die Kommunikationslücke in Projekten neu überbrücken müssen. Dies bedeutet, sie müssen ein gemeinsames Verständnis erarbeiten.

Die initiale Erhebung des Problems obliegt nicht immer dem Auftragnehmer, in der von uns durchgeführten Studie ist dies aber der Fall.

In der Praxis sind aufbauorganisatorische Beschreibungen häufig vorhanden, Prozessbeschreibungen sind hingegen weniger häufig explizit vorhanden. Sehr wohl aber gibt es typischerweise Arbeits- und Bereichsbeschreibungen, die die Arbeitsverantwortlichkeit von Personen bzw. von organisatorischen Einheiten beschreiben. In diesen sind Geschäftsprozesse häufig explizit in Form von Aufgaben genannt, aber nicht präzise genug vorhanden, um Geschäftsprozesse abzuleiten.

Ein Softwaresystem, eingesetzt in einem Unternehmen das auf Geschäftsprozesse ausgerichtet ist (ob explizit oder implizit), unterstützt eine Reihe von Geschäftsprozessen. Alle diese Geschäftsprozesse müssen für das Erarbeiten eines Problemverständnisses so weit verstanden werden, dass der Zusammenhang zwischen diesen und den darin involvierten Geschäftsobjekten (z. B. Rechnung, Lieferschein, usw.) bekannt ist und die Art und Weise der Unterstützung von Geschäftsprozessen erhoben wurde.

Um dieses Problemverständnis zu erheben, werden in frühen Phasen des Requirements Engineerings in der Praxis eine Vielzahl von Artefakten erstellt. Dazu werden z. B. durch Beobachtungen Abläufe expliziert, in Interviews Wissen erfragt, usw.

Die Inhalte in diesen Artefakten werden häufig so in der Abstraktionsebene erfasst, wie sie von den befragten Personen preisgegeben werden. Je nach Ausschnitt und

der Fähigkeit des Erhebers (z. B. im Falle eines speziellen Analysten), wird durch Nachfrage mehr oder weniger Wissen zusätzlich erfasst.

Diese Modelle schaffen häufig kein Gesamtverständnis über eine Lösung, sondern detaillieren Wissensbereiche, deren Zusammenhang häufig nicht klar gegeben sind. Die Artefakte sind daher losgelöst voneinander. Viele der Probleme, die sich bezüglich eines Zusammenhangs ergeben, ergeben sich nicht im Requirements Engineering, sondern häufig erst in späteren Phasen, wenn dieser Zusammenhang präzisiert werden muss (sofern er überhaupt bekannt ist).

Zu dem kommt noch ein zusätzliches Problem hinzu: Softwareentwickler kennen häufig nur ihren über Schnittstellen spezifizierten Arbeitsbereich und die Anforderungen, die darin umzusetzen sind. Dies ist dann kein Problem, wenn die Schnittstellenspezifikation exakt genug ist. Häufig ist dies in der Softwareentwicklung aber nicht der Fall. Das von Selic [Sel12] angesprochene Fehlen eines Gesamtverständnisses über das System in der Entwicklung führt dazu, dass diese Fehler auch von den Softwareentwicklern nicht einfach gefunden werden, sondern sich dieses Problem erst bei der Integration zeigt. Damit ist die Entdeckung noch später gegeben, was zu erheblichen Mehrkosten im Softwareprojekt führt.

Die voneinander losgelösten Artefakte unterstützen das Einarbeiten in einen Problembereich bzw. auch in den Lösungsbereich nicht, da relevantes Wissen nicht einfach identifizierbar ist.

Auf Seiten des Auftraggebers kommt es zu ähnlichen Problemen mit den Artefakten, die vom Auftragnehmer erstellt wurden. Die Artefakte sind häufig nicht verständlich für die Stakeholder des Auftraggebers. Das zu erstellende System, z. B. präzisiert im Lastenheft, kann daher nicht angemessen validiert werden. Dies führt in späteren Phasen zu erheblichen Kostenabweichungen.

In der Literatur. In der Literatur wird dieses Problem u.a. von Curtis et al. [CKI88] erwähnt. Curtis et al. [CKI88] haben eine Studie durchgeführt: Es wurden u.a. Softwareentwickler befragt, wobei eine wichtige Aussage jene war, dass eines der Kernprobleme in der Softwareentwicklung das Fehlen von Systemebenen-Denker (org. *System-level thinkers*) ist, welche die Arbeiten jener koordinieren, die (wie in diesem Abschnitt oben erwähnt), nur einen Ausschnitt des Systems betrachten. In der Studie von Curtis et al. [CKI88] wird erläutert, wie wichtig diese Sichtweise für die Softwareentwicklung allgemein und die involvierten Softwareentwickler ist.

5.8.2 Auswirkungen

Die Auswirkungen dieses Problems sind in der Literatur vielfach beschrieben, ebenso ist die Relevanz dieses Problems unumstritten.

Die Auswirkungen wurden in Abschnitt 5.8.1 bereits angedeutet.

5.8.3 Mögliche Lösung des Problems

Ein optimaler Zustand wäre dann erreicht, wenn sich alle Stakeholder ein gemeinsames Verständnis teilen.

Lösungsidee. Wir sind uns bewusst, dass wir zu diesem Problem nur einen kleinen Beitrag zur Lösung leisten können. Dieses Erarbeiten eines stakeholderweiten Verständnisses ist auch Teilgebiet des Wissensmanagements und vieler weiteren Forschungsrichtungen.

Die Erhebung von Wissen über den Problembereich entspricht einem Monolog auf Seiten des Auftraggebers. Z. B. wird in Interviews Wissen erhoben, teilweise Detailfragen gestellt, um Ungereimtheiten zu klären. Es bleibt aber ein Monolog. Dies ist für die Erhebung von Details durchaus sinnvoll, aber bezüglich zur Erhebung eines gemeinsamen Verständnisses sehen wir das nicht so.

Wir sehen die Erhebung eines initialen gemeinsamen Verständnisses zwischen Auftraggeber und Auftragnehmer als Dialog. Ziel ist die Entwicklung eines Systems, das den Problembereich unterstützt.

Unsere Lösungsidee besteht darin, die Erfassung dieses initialen Verständnisses einer Vorgehensweise folgen zu lassen, die einen Dialog darstellt und abwechselnd Ausschnitte des Problems- und Lösungsbereiches behandelt.

Hierzu besteht die Vorgehensweise aus Schritten, die jeweils vom Auftraggeber oder vom Auftragnehmer durchzuführen sind, wobei jeder Schritt eine Menge von Fragen enthält, die in innerhalb eines Schrittes zu beantworten sind. Die Schritte basieren auf den gefundenen Inhalte in Modellen, erstellt in frühen Phasen des Requirements Engineerings (siehe Kapitel 4). Auf Basis der Antworten wird ein Modell (das Basismodell) erstellt. Dieses Basismodell repräsentiert das gemeinsame Verständnis. Aufbauend auf das Basismodell werden alle weiteren Modelle in frühen Phasen des Requirements Engineering erstellt.

5.8.4 Kriterium

Wir sehen das Problem dann als gelöst, wenn folgendes Kriterium erfüllt ist:

- **K7.1:** Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.

5.9 Zusammenfassung

Dieses Kapitel beschreibt die Problemstellung dieser Arbeit in Hinblick auf eine Lösung sowie lösungsrelevante Kriterien.

Dazu wurden die Erkenntnisse aus Kapitel 3 und 4 zusammengeführt. Zu jedem Problem wurden die Lösungsideen präsentiert und der Optimalzustand beschrieben. Die beschriebenen Optimalzustände dienen als Kriterien zur Evaluierung des Ansatzes.

Ontologische Basis

Dieses Kapitel führt in die Modellierungssprache des Ansatzes ein.

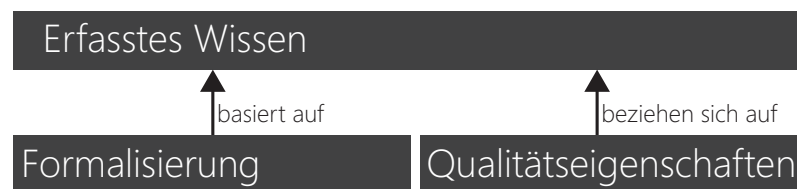


Abbildung 6.1: Aufbau des Kapitels

Die in Kapitel 7 näher beschriebene Vorgehensweise basiert auf der ontologischen Basis. Das erstellte Modell nennen wir Basismodell. Der Zweck des Basismodell ist es, (a) inhaltliche Missverständnisse zu reduzieren, (b) als Basis für andere Artefakte dafür zu sorgen, dass sich diese Artefakte auf einer ähnlichen Abstraktionsebene befinden, (c) die Konsistenz der darauf basierenden Artefakte untereinander zu verbessern, (d) die Modellbildung so zu lenken, dass ähnliche Sachverhalte auch ähnlich repräsentiert werden, (e) eine ausgeglichene Repräsentation von Wissen in allen auf dem Basismodell basierenden Artefakten zu erreichen, (f) die Möglichkeit der Verknüpfung zu anderen Artefakt-Typen zu bieten und (g) ein stakeholderweites Verständnis zu repräsentieren.

Definition ontologisches Konzept:

Ein ontologisches Konzept (in dieser Arbeit kurz Konzept) beschreibt eine Menge von Eigenschaften, die von einer oder mehreren Entitäten erfüllt sein können. Ein ontologisches Konzept fasst daher Eigenschaften zusammen und entspricht einer Vorstellung von etwas, bei dem die Eigenschaften bekannt sind.

Definition Entität:

Eine Entität bezeichnet einen konkreten oder abstrakten Gegenstand. Im Kontext dieser Arbeit sprechen wir auch von einer Entität, wenn ein Begriff durch die Zuordnung zu einem Konzept stereotypisiert wurde.

Abbildung 6.1 zeigt den Aufbau des Kapitels. In Abschnitt 6.2 führen wir ein Beispiel-Basismodell ein. In Abschnitt 6.2 (Erfasstes Wissen) beschreiben wir informal

jene Art von Wissen, die mit Hilfe der ontologische Basis erfasst werden kann. In Abschnitt 6.3 (Formalisierung) formalisieren wir das zuvor informal Beschriebene. Abschließend werden in Abschnitt 6.4 (Qualitätseigenschaften eines Basismodells) die Qualitätseigenschaften eines Basismodells expliziert.

6.1 Beispiel

Dieser Abschnitt zeigt ein Produkt des Modellierungsansatzes. Abbildung 6.2 zeigt ein Basismodell. Es inkludiert die Struktur von Aufbau- und Ablauforganisation einer Unternehmung, sowie die involvierten Geschäftsobjekte mit ihrer Art der Involvierung. Zusätzlich wird Wissen über das System in der Entwicklung erfasst. Softwareseitig wird das System in Abbildung 6.2 als Unterstützung des Geschäftes dargestellt. Geschäftsobjekte und teilweise auch Personen bzw. Rollen werden softwareseitig in Struktur abgebildet.

Diese Trennung zwischen Problem- und Lösungsdomäne ermöglicht es uns, in frühen Phasen des Requirements Engineerings, die in andere Modellierungstechniken beschriebenen Inhalte (durch die explizite Nennung der Entitäten mit expliziter Zugehörigkeit zu einer Problemdomäne oder zu einer Lösungsdomäne), (a) einer Domäne zuordnen zu können und (b) präzisere Aussagen über die modellierten Inhalte zu treffen.

Das in Abbildung 6.2 eingeführte Beispiel-Basismodell wird auch in Kapitel 7 verwendet, um die Verknüpfung zu anderen Artefakt-Typen zu beschreiben.

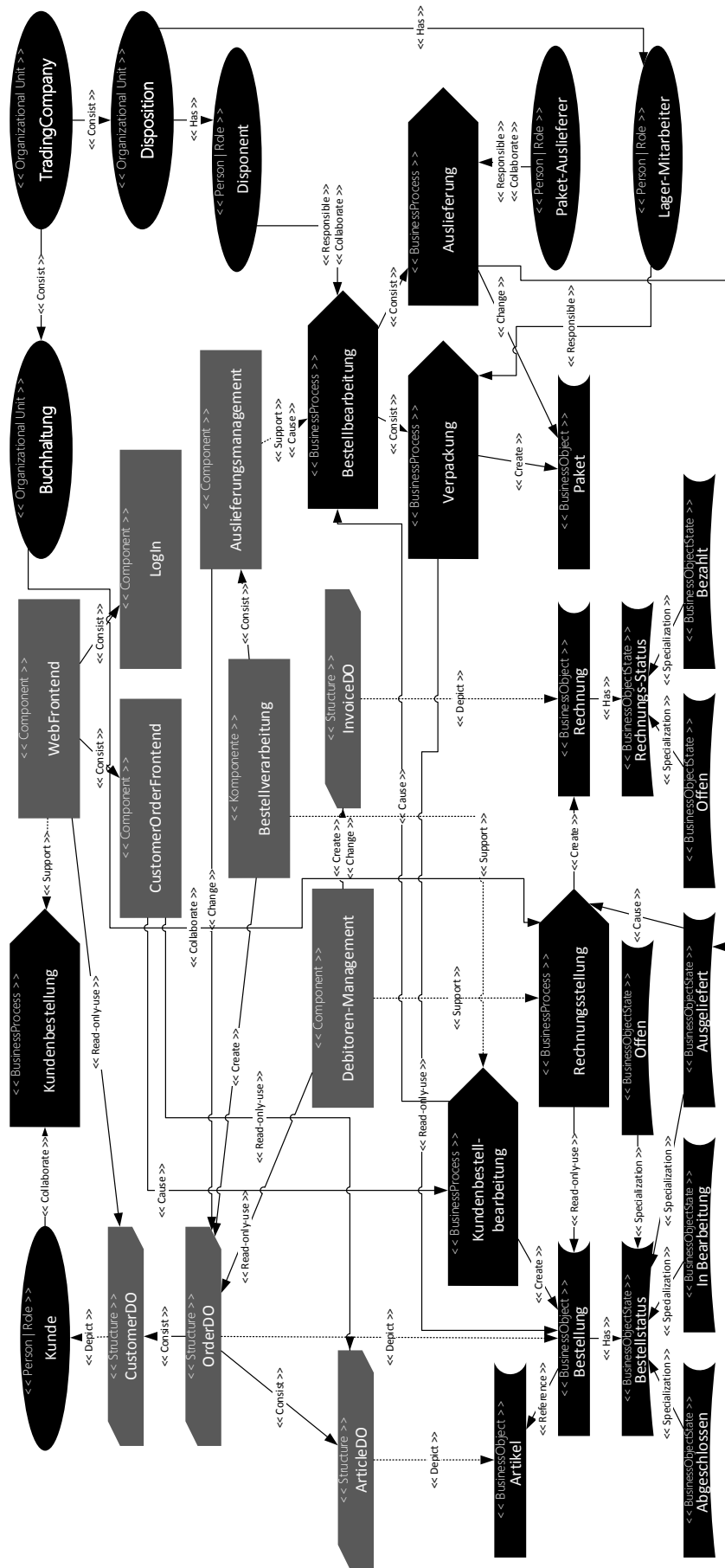


Abbildung 6.2: Beispiel eines Basismodells. Das nicht vollständige Beispiel zeigt das erhobene Gesamtverständnis einer Unternehmung, die Bestellungen über das Internet annimmt, bearbeitet und ausliefert.

6.2 Erfasstes Wissen

Basierend auf den Inhalten die in Artefakten erfasst wurden, welche in frühen Phasen des Requirements Engineering erstellt wurden, legen wir in diesem Abschnitt jene Arten von Wissen dar, die wir im Rahmen der Erstellung von Basismodellen erfassen. Hierzu werden alle Konzepte und Beziehungen der ontologischen Basis erläutert.

Zur Entstehung der ontologischen Basis. Die Terminologie der ontologischen Basis basiert auf der Enterprise Ontologie von Uschold et al. [UKMZ98]. Die Inhalte von Artefakten, erstellt in frühen Phasen des Requirements Engineerings (siehe Kapitel 4), wurden den Konzepten und Beziehungen der Enterprise Ontologie zugeordnet. Jene Konzepte und Beziehungen die dabei übrig geblieben sind, bilden die initiale Terminologie der ontologischen Basis. Übrig geblieben sind Konzepte und Beziehungen dann, wenn Aussagen in Artefakten Konzepten bzw. Beziehungen der Enterprise Ontologie zugeordnet werden konnten. Einige Beziehungen, die in Artefakten häufig vorkamen aber nicht mit Hilfe der Enterprise Ontologie ausgedrückt werden konnten, wurden hinzugefügt (dies gilt insbesondere für die Unterteilung der Arten der Involvierung von Geschäftsobjekten in Geschäftsprozessen durch die Verwendung der Arbeiten von Dori [Dor11]).

Das Basismodell ist das Produkt des Konstruktionsprozesses (siehe Kapitel 7). Das explizierte Wissen im Basismodell basiert auf Konzepten und Beziehungen in der ontologischen Basis.

Abbildung 6.3 zeigt den Zusammenhang zwischen wichtigen Begriffen des Modellierungsansatzes. Begriffe werden mit Hilfe der Kompetenzfragen, siehe dazu Abschnitt 7.3, erhoben. Mit Hilfe der Kompetenzfragen wird bereits ein erstes Scoping¹ durchgeführt.

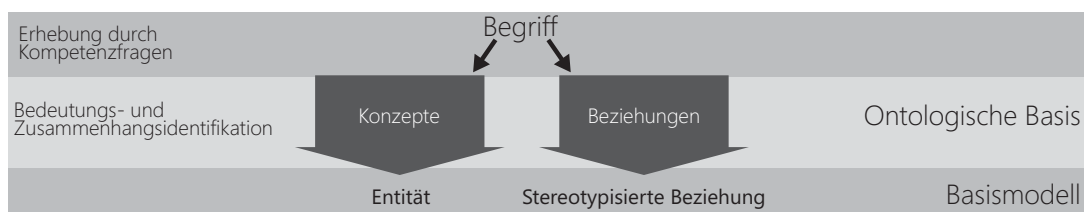


Abbildung 6.3: Zusammenhang der Begriffe

Mit Hilfe der Konzepte und Beziehungen in der ontologischen Basis werden die Bedeutung von Begriffen und mögliche Zusammenhänge mit anderen Begriffen (die wir dann Entitäten nennen) identifiziert. Wir sprechen dann von einer Entität, wenn ein Begriff sich die Eigenschaften genau eines Konzeptes teilt und wir sprechen dann von stereotypisierter Beziehung, wenn sich ein Begriff die Eigenschaften genau einer Beziehung aus der ontologische Basis teilt, die als Stereotyp für diese stereotypisierte Beziehung dient. Ein Begriff ist entweder als Entität oder als stereotypisierte Beziehung in einem Basismodell repräsentiert.

¹ Scoping leitet sich aus dem Englischen (*scope*) ab und bedeutet soviel wie die Festlegung eines Geltungsbereiches bzw. das Abgrenzen eines Bereiches.

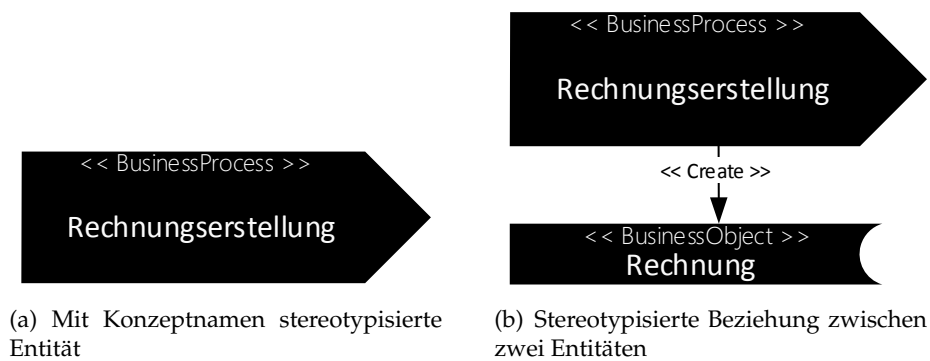


Abbildung 6.4: Beispiel der Verwendung von Konzepten und Beziehungen als Stereotypen von Entitäten und deren Beziehungen

Definition stereotypisierte Beziehung:

Eine stereotypisierte Beziehung ist eine Beziehung zwischen zwei Entitäten, deren Bedeutung durch die Übereinstimmung von Eigenschaften genau einer möglichen Beziehung aus der ontologische Basis identifiziert wurde.

Die Beziehungen zwischen den Konzepten stammen initial aus den Artefakten, die in frühen Phasen des Requirements Engineerings erstellt wurden. Die gefundenen Konzepte und deren Beziehungen zueinander wurden dann als Basis für die Erstellung von Basismodellen in verschiedenen persönlichen Entwicklungsprojekten und Entwicklungsprojekten von Studenten verwendet. Das Feedback wurde wieder zurück in den Ansatz integriert. Dieser Vorgang wurde über drei Semester wiederholt. Der Ansatz wurde dann in der Industrie eingesetzt. Das Feedback wurde wieder in den Ansatz integriert.

Im Ansatz ist jede Entität genau einem ontologischem Konzept zugehörig. Eine Zugehörigkeit wird durch die Angabe des Konzeptes als Stereotyp angegeben. Dies wird in Abbildung 6.4 in Teil (a) gezeigt.

Im Modellierungsansatz werden nur Beziehungen zwischen Entitäten erlaubt, die durch die ontologische Basis gegeben sind. Teil (b) der Abbildung 6.4 zeigt eine Beziehung zwischen den Entitäten *Rechnungserstellung* und *Rechnung*. Die Beziehung weist den Stereotyp *Create* auf. Wir sprechen daher bei dieser Beziehung von einer stereotypisierten Beziehung.

6.2.1 Konzepte der Problemdomäne

• Konzept: **BusinessProcess**

Ein Geschäftsprozess (*BusinessProcess*) besteht aus einer Menge von auszuführenden Aktivitäten. Geschäftsprozesse können im Ansatz z. B. Kernprozesse² sein, aber auch unterstützende Prozesse (z. B. Zulieferprozesse, bzw. Vorbereitungs- und Nachbereitungsprozesse) sein.

² Darunter verstehen wir eine Verknüpfung von zusammenhängenden Aktivitäten, Entscheidungen, Informationen und Materialflüssen, die zusammen den Wettbewerbsvorteil eines Unternehmens ausmachen [OF06].

Der Geschäftsprozess wird in der Enterprise Ontologie von Uschold et al. [UKMZ98] *Activity* genannt, die Eigenschaften entsprechen denen von Geschäftsprozessen. Wir verwenden daher in dieser Arbeit den gebräuchlicheren Namen *Geschäftsprozess* bzw. in Modellen *BusinessProcess*.

Geschäftsprozesse können aus anderen Geschäftsprozessen bestehen und sich gegenseitig anstoßen.

- Konzept: **BusinessObject**

Entspricht einem existierenden (oder zumindest einmal im Kontext des dargestellten Problemdomäne existierenden) materiellen oder immateriellen, konkreten oder abstrakten Gegenstand, der im Rahmen eines Geschäftsprozesses verwendet, bearbeitet, erstellt bzw. eliminiert werden kann. In der Literatur werden diese Geschäftsobjekte (*BusinessObjects*) auch Daten genannt [All09, RF10].

In der Enterprise Ontologie von Uschold et al. [UKMZ98] werden die Geschäftsobjekte *Resources* genannt. Sie entsprechen *Daten, Gegenständen, usw.* aus anderen Modellierungstechniken und teilen sich die Eigenschaft der Involviertheit in Geschäftsprozesse. Wir nennen diese *Resources* Geschäftsobjekte.

- Konzept: **BusinessObjectState**

Bezeichnet den Zustand eines Geschäftsobjektes (*BusinessObjectState*). Der Zustand entspricht z. B. einem Bearbeitungszustand eines Geschäftsobjektes.

Uschold et al. [UKMZ98] führen für die Änderungen an Geschäftsobjekten Effekte ein. Daher hat die Ausführung eines Geschäftsprozesses Effekte. Wir stellen diese Effekte über Zustände dar, da diese einfacher zu modellieren sind und keine eigenen Entitäten benötigen, um die Auswirkung eines Effektes genauer zu beschreiben. In den erhobenen Inhalten aus Artefakten, erstellt in frühen Phasen des Requirements Engineerings, werden Zustände häufig in Modellen erfasst. Die Zustände darin entsprechen eher der Wahrnehmung als einer Auswirkung im Rahmen eines Geschäftsprozesses.

- Konzept: **Person | Role**

Eine Person oder Rolle (*Role*) bezeichnet eine konkrete Person oder deren fachliche Funktion im Rahmen der dargestellten Domäne.

Uschold et al. [UKMZ98] führt kein Konzept *Role* in der Enterprise Ontologie ein. Wir haben in Zusammenarbeit mit den Unternehmen festgestellt, dass die Unterscheidung zwischen Rolle und Person häufig nicht möglich ist. Z. B. eine Entität trägt den Namen *Buchhalter*. Wird die Entität nicht genauer beschrieben bleibt offen, ob es sich beim Buchhalter um eine Person oder Rolle handelt. Wenn wir von einer Rolle sprechen, geht dies typischerweise mit Erwartungen in Form von Verantwortung oder Mitarbeiter in einem Geschäftsprozessen einher. Da wir die Verantwortung und Mitarbeit einer Person bzw. einer Rolle im Modellierungsansatz explizit modellieren, modellieren wir auch diese Erwartungen. Ob die Verantwortung von einer konkreten Person oder von einer Rolle erfüllt wird, ist für das zu entwickelnde Softwaresystem nicht von Bedeutung. Wir haben uns daher entschlossen hier keine Unterscheidung zu machen und fügen dem in der Enterprise Ontologie vorhandenen Konzept *Person* den Zusatz *Role* hinzu.

- Konzept: **Maschine**

Eine Maschine ist ein Akteur, der ein externes System darstellt. Eine Maschine ist in der Lage einen Geschäftsprozess zu überwachen, bzw. Aktivitäten eines Geschäftsprozesses auszuführen.

Abgrenzung zur Rolle einer Maschine: Eine Maschine spielt in unserem Kontext sehr wohl eine Rolle in z. B. Geschäftsprozessen. Wir sehen die Rolle im Kontext der Modellierung eines Basismodells als eine dem Menschen vorbehaltenes Verhaltensmuster, das erlernt und von einer Person in einer bestimmten Situation ausgeführt werden kann [Bro02].

- Konzept: **Organizational Unit**

Eine organisatorisch Einheit bzw. Organisationseinheit (*Organizational Unit*) ist eine Einheit, die Gruppen von Personen bzw. Rollen, Maschinen oder andere organisatorische Einheiten zusammenfasst. Z. B. ist die Buchhaltung eine organisatorische Einheit. Sie fasst andere organisatorische Einheiten (z. B. Debitorenbuchhaltung), aber auch Personen bzw. Rollen (z. B. Buchhalter) zusammen.

6.2.2 Beziehungen zwischen Konzepten der Problemdomäne

- Beziehung: **BusinessProcess –read-only-use→ BusinessObject**

Im Rahmen der Ausführung eines Geschäftsprozesses wird ein Geschäftsobjekt verwendet, ohne dieses zu ändern (*read-only-use*).

Die Art der Involviertheit eines Geschäftsobjektes in einen Geschäftsprozess (*read-only-use, create, ...*) stammt aus der Erhebung der Inhalte von Artefakten, erstellt in frühen Phasen des Requirements Engineerings, siehe dazu Kapitel 4.

- Beziehung: **BusinessProcess –create→ BusinessObject**

Im Rahmen der Ausführung eines Geschäftsprozesses wird ein Geschäftsobjekt erstellt (*create*).

- Beziehung: **BusinessProcess –change→ BusinessObject**

Im Rahmen der Ausführung eines Geschäftsprozesses wird ein Geschäftsobjekt geändert (*change*).

- Beziehung: **BusinessProcess –eliminate→ BusinessObject**

Im Rahmen der Ausführung eines Geschäftsprozesses wird ein Geschäftsobjekt eliminiert (*eliminate*).

- Beziehung: **BusinessProcess –cause→ BusinessProcess**

Im Rahmen der Ausführung eines Geschäftsprozesses wird ein anderer oder der gleiche Geschäftsprozess ausgelöst (*cause*). Mit seiner Auslösung geht die Ausführung einher.

- Beziehung: **BusinessProcess –consist→ BusinessProcess**

Ein Geschäftsprozess besteht aus einem anderen Geschäftsprozess. Diese Beziehung erlaubt uns das Zusammenfügen von Geschäftsprozessen zu *größeren* Geschäftsprozessen. Zusammengefügte Geschäftsprozesse vererben bestimmte Beziehungen an

übergeordnete Geschäftsprozesse weiter. Siehe dazu Abschnitt 6.3.

- Beziehung: **BusinessObject –consist→ BusinessObject**

Ein Geschäftsobjekt besteht (*consist*) aus einem anderen Geschäftsobjekt. Diese Beziehung erlaubt die Hierarchisierung von Geschäftsobjekten.

- Beziehung: **BusinessObject –reference→ BusinessObject**

Ein Geschäftsobjekt referenziert (*reference*) ein anderes Geschäftsobjekt. Eine Referenz drückt eine Beziehung aus, die sich nicht physikalisch manifestiert. Z. B. referenziert eine Bestellung als Geschäftsobjekt (z. B. in Form eines Ausdruckes) Artikel. Die Bestellung besteht aber nicht aus Artikeln, sondern referenziert diese nur.

- Beziehung: **BusinessObject –has→ BusinessObjectState**

Ein Geschäftsobjekt hat (*has*) einen Zustand. Der Zustand drückt z. B. den Bearbeitungsstand eines Geschäftsobjektes aus und spiegelt daher einen inneren Zustand eines Geschäftsobjektes wieder.

- Beziehung: **BusinessObjectState –cause→ BusinessProcess**

Die Änderung eines Zustandes stößt einen Geschäftsprozess an (*cause*).

Die Auslösung eines Geschäftsprozesses in der Enterprise Ontologie [UKMZ98] wird u.a. über den Effekt vorgenommen. Wie bereits zum Thema Zustand eines Geschäftsobjektes argumentiert, verzichten wir auf die Einführung einer zusätzlichen Entität und drücken diesen Effekt über die *cause*-Beziehung aus, die dem Zusammenhang aus Effekt-Beschreibung und Ausführung eines Geschäftsprozesses aus der Enterprise Ontologie entspricht.

- Beziehung: **BusinessObjectState –cause→ BusinessObjectState**

Die Änderung eines Zustandes löst die Änderung eines anderen Zustandes aus. Diese Beziehung wird dann verwendet, wenn eine Zustandsänderung einen anderen Zustand verursacht, dessen Zustandekommen keiner Aktivitäten im Rahmen von Geschäftsprozessen bedarf. Z. B. kann die Änderung eines Zustandes im Rahmen der Disposition einer Bestellung (z. B. *zusammengestellt*) den Zustand *versandfertig* verursachen.

- Beziehung: **BusinessObjectState –specialize→ BusinessObjectState**

Ein Zustand eines Geschäftsobjektes spezialisiert (*specialize*) dann einen anderen Zustand desselben Geschäftsobjektes, wenn dieser den anfänglichen Zustand weiter detailliert, z. B. ein Zustand *Lieferstatus* kann von einem Zustand *Ausgeliefert* spezialisiert werden.

- Beziehung: **Maschine –responsible→ BusinessProcess**

Eine Maschine überwacht die Ausführung von Aktivitäten eines Geschäftsprozesses. Wir sprechen von Verantwortlichkeit im Zusammenhang einer Maschine mit einem Geschäftsprozess, wenn eine Überwachung gegeben ist, die eine Ausführung der Aktivitäten eines Geschäftsprozesses nicht impliziert.

- Beziehung: **Maschine –collaborate→ BusinessProcess**

Eine Maschine führt Aktivitäten eines Geschäftsprozesses aus (*colaborate*). Werden Geschäftsprozesse von Maschinen ausgeführt, spricht man auch von Workflow

[Mel08].

- Beziehung: **OrganizationalUnit –collaborate→ BusinessProcess**

Eine organisatorische Einheit arbeitet in den Aktivitäten eines Geschäftsprozesses mit.

- Beziehung: **OrganizationalUnit –responsible→ BusinessProcess**

Eine organisatorische Einheit ist für einen Geschäftsprozess verantwortlich.

- Beziehung: **Person | Role –collaborate→ BusinessProcess**

Eine Person oder Rolle arbeitet in den Aktivitäten eines Geschäftsprozesses mit.

- Beziehung: **Person | Role –responsible→ BusinessProcess**

Eine Person oder Rolle ist für einen Geschäftsprozess verantwortlich. Die Verantwortlichkeit impliziert keine Mitarbeiter an den Aktivitäten eines Geschäftsprozesses.

- Beziehung: **Maschine –consist→ Maschine**

Eine Maschine besteht aus (*consist*) einer anderen Maschine. Z. B. CRM-System besteht aus einem Kundenverwaltungssystem.

- Beziehung: **Organizational Unit –consist→ Organizational Unit**

Eine organisatorische Einheit besteht aus anderen organisatorischen Einheiten. Z. B. eine Buchhaltung besteht aus einer Debitorenbuchhaltung.

- Beziehung: **Organizational Unit –has→ Maschine**

Eine Maschine gehört zu einer organisatorischen Einheit. Z. B. das Finanzamt hat ein Umsatzsteuerabfragesystem.

- Beziehung: **Organizational Unit –has→ Person | Role**

Eine organisatorische Einheit hat Personen oder Rollen, die der organisatorischen Einheit zugeordnet sind.

6.2.3 Allgemeine Konzepte

- Konzept: **Quality**

Bezeichnet eine Qualitätseigenschaft, die im Rahmen des Softwareprojektes definiert wurde (z. B. Usability-Eigenschaften).

- Konzept: **Constraint**

Bezeichnet eine Bindung, die nicht innerhalb des Softwareprojektes definiert wurde, aber deren Erfüllung zwangsweise gegeben sein muss (z. B. gesetzliche Vorgaben, physikalische Gesetze, ...)

Quality und Constraint werden in der Enterprise Ontologie [UKMZ98] über Spezifikations-Entitäten ausgedrückt. In der Erhebung der Inhalte von Artefakte, erstellt in frühen Phasen des Requirements Engineerings, haben wir eine Unterscheidung festgestellt, die wir übernommen haben.

6.2.4 Beziehungen zwischen den allgemeinen Konzepten und den Konzepten der Problem- und Lösungsdomäne

- Beziehung: ? –**fulfill**→ **Quality**

Ein Entität erfüllt (*fulfill*) eine Qualitätseigenschaft (*Quality*), die im Rahmen des Softwareprojektes definiert wurde.

- Beziehung: ? –**fulfill**→ **Constraint**

Ein Entität erfüllt eine Bedingung (*Constraint*), deren Definition außerhalb des Projektes stattfand und im Rahmen des Softwareprojektes zu erfüllen ist.

Wobei das „?“ für *BusinessProcess*, *BusinessObject*, *Structure* oder *Component* stehen kann.

6.2.5 Konzepte der Lösungsdomäne

In frühen Phasen der Softwareentwicklung werden, wenn Softwareanteile erfasst werden, Einheiten von Funktionen in einer Black-Box-Sicht erfasst. Diese Einheiten von Funktionen führen Aktivitäten aus. Diese Aktivitäten werden teilweise benannt und entsprechen Geschäftsprozessen. Siehe dazu Kapitel 5. Wir nennen diese Black-Box-Sicht Komponenten bzw. *Components*.

- Konzept: **Component**

Eine Komponente ist ein verwendbares in sich abgeschlossenes Softwareartefakt, dessen Schnittstellen wohl definiert und deren Realisierung verborgen ist und auch in Kombination mit anderen Komponenten eingesetzt werden kann [ABC⁺02].

- Konzept: **Structure**

Eine Struktur ist eine auf seine Verwendung hin abstrahiertes Abbildung seines Originals aus der Problemdomäne. Eine Struktur bildet die softwarerelevanten strukturellen Eigenschaften eines Originals aus der Problemdomäne ab.

6.2.6 Beziehungen zwischen Konzepten der Lösungsdomäne

- Beziehung: **Component** –**consist**→ **Component**

Eine Komponente kann aus anderen Komponenten zusammengesetzt sein. Diese Beziehung erlaubt es, mehrere Komponenten zu einer *größeren* Komponente zusammenzusetzen. Wir sprechen in diesem Zusammenhang auch von der Komposition von Komponenten.

Werden Komponenten zu einer übergeordneten Komponente zusammengesetzt, so vererben diese bestimmte Beziehungen an die übergeordnete Komponente. Siehe dazu Abschnitt 6.3.

- Beziehung: **Component** –**require**→ **Component**

Eine Komponente benötigt (*require*) eine andere Komponente.

In der Formalisierung in Abschnitt 6.3 schließen wir u.a. dann auf eine *require*-Beziehung zwischen zwei Komponenten (*c1* und *c2*), wenn *c2* eine Instanz einer Struktur

in der Ausführung der Funktionen der Komponente involviert, die Instanz aber von der Komponente *c1* erzeugt wird.

In der Zusammenarbeit mit der Industrie haben wir festgestellt, dass die Involviertheit von Instanzen von Strukturen in Komponenten häufig von der Involviertheit der Geschäftsobjekte in Geschäftsprozessen abweicht (z. B. aufgrund von Anforderungen an das Softwaresystem). Deshalb haben wir die zwischen Geschäftsprozessen und Geschäftsobjekten identifizierten Beziehungen auch für die Beziehungen zwischen Komponenten und Strukturen eingeführt.

- Beziehung: **Component –read-only-use→ Structure**

Eine Komponente verwendet eine Instanz einer Struktur lesend, ohne diese zu ändern.

- Beziehung: **Component –create→ Structure**

Eine Komponente erstellt eine Instanz einer Struktur.

- Beziehung: **Component –change→ Structure**

Eine Komponente ändert die Instanz einer Struktur.

- Beziehung: **Component –eliminate→ Structure**

Eine Komponente eliminiert die Instanz einer Struktur.

Wie in den Inhalten in Artefakten erhoben (siehe Kapitel 4), stehen die Entitäten in Beziehung untereinander, wir haben dafür *consist* als Komposition und *reference* als Aggregation bzw. einfache Referenz gewählt.

- Beziehung: **Structure –consist→ Structure**

Eine Struktur besteht aus einer anderen Struktur. Diese Beziehung erlaubt uns das Zusammensetzen von Strukturen zu *größeren* Strukturen.

- Beziehung: **Structure –reference→ Structure**

Eine Struktur referenziert eine andere Struktur.

Z. B. besteht (*consist*) eine Struktur *Kunde* u.a. aus einer Struktur *Adresse*. Eine Struktur *Bestellung* besteht aber nicht aus der Struktur *Kunde*, sondern referenziert diese nur, um auszudrücken, dass es einen Zusammenhang zwischen Bestellung und Kunde gibt.

6.2.7 Beziehungen zwischen Konzepten der Problem- und Lösungsdomäne

Um die Beziehung zwischen der Lösungsdomäne und der Problemdomäne herzustellen, bedienen wir uns den Arbeiten die im Rahmen der Geschäftsprozessmodellierung getätigt wurden, z. B. wird in der Business Process Model and Notation³ Software über so genannte *data types* involviert. Domänenexperten in der von uns betrachteten Problemdomäne sind mit diesen Notationen vertraut, deswegen involvieren wir Strukturen bzw. *Structure*, stellvertretend für diese *data types*, indem wir

³ Business Process Model and Notation (BPMN) ist eine Sprache zur Modellierung von Geschäftsprozessen und Arbeitsabläufen der Object Management Group (OMG) [Obj06].

sie explizit über *depict* in der Lösungsdomäne abbilden und die Beziehung zwischen den Bereichen so explizieren.

- Beziehung: **Structure –depict→ Maschine**

Eine Struktur bildet eine Maschine softwareseitig ab.

- Beziehung: **Structure –depict→ Person | Role**

Eine Struktur bildet eine Person oder Rolle softwareseitig ab.

- Beziehung: **Structure –depict→ OrganizationalUnit**

Eine Struktur bildet eine organisatorische Einheit softwareseitig ab.

- Beziehung: **Structure –depict→ BusinessObject**

Eine Struktur bildet ein Geschäftsobjekt softwareseitig ab.

- Beziehung: **Structure –depict→ BusinessObjectState**

Eine Struktur bildet einen Zustand eines Geschäftsobjektes softwareseitig ab.

In Notationen um Geschäftsprozesse zu modellieren, wie z. B. erweiterte ereignisgesteuerte Prozessketten⁴, übernimmt Software eine unterstützende Funktion. Diese unterstützende Funktion einer Software bilden wir über die Beziehung *support* ab.

- Beziehung: **Component –support→ BusinessProcess**

Eine Komponente unterstützt einen Geschäftsprozess. Dies ist dann der Fall, wenn der Geschäftsprozess Aktivitäten beinhaltet, die mit Hilfe des Softwaresystems (dargestellt durch die Komponente) durchgeführt werden.

- Beziehung: **Component –cause→ BusinessProcess**

Eine Komponente stößt einen Geschäftsprozess an. Dies ist dann der Fall, wenn z. B. im Rahmen einer Ausführung einer Funktion einer Komponente ein Geschäftsprozess angestoßen wird.

6.3 Formalisierung

In diesem Abschnitt werden alle Konzepte und Beziehungen mittels algebraischer Spezifikation formalisiert. Die Erläuterung zu den Konzepten und Beziehungen ist in Abschnitt 6.2 zu finden.

Zur Herkunft der Axiome. Alle in diesem Abschnitt zu findenden Axiome stammen aus mindestens einer der folgenden Quellen: (a) Aus Fehlern, die in der Anwendung aufgetreten sind und wir im Basismodellen gefunden haben, (b) aus Rückmeldungen von Anwendern über mögliche Konsistenzprobleme, (c) von Kollegen am Lehrstuhl, die sich mit dem Ansatz beschäftigt haben.

⁴ Erweiterte ereignisgesteuerte Prozessketten (EPPK) ist eine Sprache zur Modellierung von Geschäftsprozessen und Arbeitsabläufen und ist wichtiges Element des ARIS-Konzeptes (Architektur Integrierter Informationssysteme) [SN00].

Algebraische Spezifikation. In algebraischen Spezifikationen wird die Beschreibung von Gegenstandstypen und deren Operationen fokussiert [BS03]. Typen werden als Namen repräsentiert. Ein Typ wird durch seine Operationen definiert. Operationen werden in Axiomen charakterisiert [BS03]. Algebraisch sind die Spezifikationen, weil die Axiome in Form von bedingten Gleichungen repräsentiert werden [BS03].

Kommentare in den Spezifikationen werden mit „//“ beginnend eingeläutet und gehen immer bis zum Ende einer Zeile.

Zur besseren Lesbarkeit kennzeichnen wir den Abschnitt mit den Operationen mit dem Schlüsselwort *operations* und den Abschnitt mit den Axiomen mit dem Schlüsselwort *axioms*.

Spezifikation der Sorte Bool. Die Sorte Bool wird zur Spezifikation von Aussagen über Beziehungen in den Formalisierungen der Konzepte verwendet.

Die Spezifikation in Listing 6.1 stammt aus [BS03]:

```

1 SPEC BOOL = {
2   sort Bool,
3   operations
4     true, false : Bool,
5     ¬           : Bool → Bool,
6     ∨, ∧, ⇒     : Bool → Bool,
7
8     Bool generated_by true, false
9   axioms
10    true ≠ false,
11    ¬ true = false,
12    ¬ ¬ x = x,
13    (false ∨ x) = (x ∨ false) = x,
14    (true ∨ x) = (x ∨ true) = true,
15    (true ∧ x) = (x ∧ true) = x,
16    (false ∧ x) = (x ∧ false) = false,
17    (x ⇒ y) = (¬ x ∨ y)
18 }

```

Listing 6.1: Spezifikation Bool

In den Spezifikationen in diesem Abschnitt wird in einigen abstrakten Datentypen die *consist*-Beziehung eingeführt. Um eine mehrfache Wiederholung der Spezifikation zu vermeiden, spezifizieren wir diese Beziehung in Listing 6.2 allgemein und verweisen an entsprechenden Stellen auf diese Teilspezifikation.

```

1 // Transitivität der consist-Beziehung
2 consist(x,y) ∧ consist(y,z) ⇒ consist(x,z),
3
4 // Antisymmetrie der consist-Beziehung
5 consist(x,y) ∧ consist(y,x) ⇒ x = y,
6
7 // Reflexivität der consist-Beziehung

```

```
8 consist(x,x)
```

Listing 6.2: Consist-Beziehung

In Listing 6.2 legen wir fest, dass es sich bei der *consist*-Beziehung um eine partielle Ordnung handelt. Es gelten daher die Eigenschaften Transitivität, Reflexivität und Antisymmetrie.

6.3.1 Problemdomäne

In diesem Abschnitt werden die Konzepte und Beziehungen der Problemdomäne spezifiziert.

```
1 SPEC BUSINESSPROCESS = {
2   sorts BusinessProcess , BusinessObject ,
3         BusinessObjectState , Bool
4   operations
5     read_only_use: BusinessProcess , BusinessObject    → Bool ,
6     create:       BusinessProcess , BusinessObject    → Bool ,
7     change:       BusinessProcess , BusinessObject    → Bool ,
8     eliminate:   BusinessProcess , BusinessObject    → Bool ,
9     cause:       BusinessProcess , BusinessProcess    → Bool ,
10    cause:       BusinessProcess , BusinessObjectState → Bool ,
11    consist:     BusinessProcess , BusinessProcess    → Bool
12  axioms
13    // Partielle Ordnung, siehe Listing 6.2
14    consist ist partielle Ordnung ,
15
16    // Exklusivität der consist-Beziehung
17    consist(x,y) ∧ consist(z,y)
18      ⇒ (consist(x,z) ∨ consist(z,x)) ,
19
20    consist(x,y) ∧ change(y,z) ⇒
21      ¬ read_only_use(x,z) ∧ change(x,z) ,
22
23    consist(x,y) ∧ create(y,z) ⇒
24      ¬ read_only_use(x,z) ∧ create(x,z) ,
25
26    consist(x,y) ∧ eliminate(y,z) ⇒
27      ¬ read_only_use(x,z) ∧ eliminate(x,z) ,
28
29    ∀ y ∈ BusinessProcess
30      (cause(x,y) ∧ consist(z,x) ⇒ cause(z,y)) ,
31
32    ∀ y ∈ BusinessObjectState
33      (cause(x,y) ∧ consist(z,x) ⇒ cause(z,y)) ,
34
35    ∀ y ∈ BusinessObjectState , ∀ z ∈ BusinessObject
36      (cause(x,y) ∧ has(z,y) ⇒
```

```

37     change(x, z) ∧ ¬ read_only_use(x, z) ,
38
39     ∀ x ∈ BusinessObject , ∃ p ∈ BusinessProcess
40     ( read_only_use(p, x) ∨ create(p, x) ∨
41       change(p, x) ∨ eliminiare(p, x) )
42 }

```

Listing 6.3: Spezifikation Geschäftsprozess

Das Axiom in Listing 6.3 in Zeile 20 stellt sicher, dass wenn ein untergeordneter Geschäftsprozess ein Geschäftsobjekt ändernd involviert, der übergeordnete Geschäftsprozess keine Beziehung zu diesem Geschäftsobjekt aufweisen darf, die eine reine lesende Involvierung aussagt und, dass der übergeordnete Geschäftsprozess dieses Geschäftsobjekt auch ändernd involviert.

Analog dazu das gleiche Vorgehen für die erstellende und eliminierende Involvierung in Zeile 23 und 26.

In Zeile 30 und 33 geben wir an, dass wenn ein untergeordneter Geschäftsprozess einen anderen Geschäftsprozess auslöst, bzw. einen Zustand eines Geschäftsobjektes verursacht, dann gilt diese Beziehung auch für den übergeordneten Geschäftsprozess.

In Zeile 35 drücken wir aus, dass wann immer ein Geschäftsprozess einen Zustand eines Geschäftsobjektes ändert, der ändernde Geschäftsprozess auch das zugrundeliegende Geschäftsobjekt ändert. Eine rein lesende Involvierung des Geschäftsobjektes ist dann nicht mehr möglich.

In Zeile 39 legen wir fest, dass ein Geschäftsobjekt immer in mindestens einen Geschäftsprozess involviert sein muss.

```

1 SPEC BUSINESSOBJECT = {
2   sorts BusinessObject , BusinessObjectState , Bool
3   operations
4     consist : BusinessObject , BusinessObject → Bool ,
5     has      : BusinessObject , BusinessObjectState → Bool
6   axioms
7     // Partielle Ordnung, siehe Listing 6.2
8     consist ist partielle Ordnung ,
9
10    // Exklusivität der consist-Beziehung
11    consist(x, y) ∧ consist(z, y)
12    ⇒ (consist(x, z) ∨ consist(z, x)) ,
13 }

```

Listing 6.4: Spezifikation Geschäftsobjekt

In Listing 6.4 wird das Geschäftsobjekt (*BusinessObject*) und in Listing 6.5 wird der Zustand eines Geschäftsobjektes (*BusinessObjectState*) spezifiziert.

```

1 SPEC BUSINESSOBJECTSTATE = {
2   sorts BusinessObjectState , BusinessObject , BusinessProcess
3   operations
4     cause : BusinessObjectState , BusinessObjectState → Bool ,

```

```

5   cause:      BusinessObjectState , BusinessProcess   → Bool ,
6   specialize: BusinessObjectState , BusinessObjectState → Bool
7 axioms
8   //Transitivität der specialize-Beziehung
9   specialize(x,y) ∧ specialize(y,z) ⇒ specialize(x,z),
10
11  //Asymmetrie der specialize-Beziehung
12  specialize(x,y) ⇒ ¬ specialize(y,x),
13
14  (specialize(x,y) ∧ x ≠ y) ∨
15    (∃ z ∈ BusinessObject (has(z,x))),
16
17  //Irreflexivität der cause-Beziehung von Zuständen
18  ¬ cause(x,x)
19 }

```

Listing 6.5: Spezifikation Zustand eines Geschäftsobjektes

In Zeile 14 legen wir fest, dass es keinen Zustand eines Geschäftsobjektes geben darf, der nicht einen anderen Zustand spezialisiert oder direkt zu einem Geschäftsobjekt gehört.

6.3.2 Lösungsdomäne

In diesem Abschnitt werden die Konzepte und Beziehungen der Lösungsdomäne spezifiziert.

```

1 SPEC COMPONENT = {
2 sorts Component, Structure , BusinessProcess , Bool
3 operations
4   consist:      Component, Component       → Bool ,
5   require:      Component, Component       → Bool ,
6   read_only_use: Component, Structure      → Bool ,
7   change:       Component, Structure       → Bool ,
8   eliminate:    Component, Structure       → Bool ,
9   create:       Component, Structure       → Bool ,
10  support:      Component, BusinessProcess → Bool ,
11  cause:        Component, BusinessProcess → Bool
12 axioms
13  //Partielle Ordnung, siehe Listing 6.2
14  consist ist partielle Ordnung,
15
16  //Exklusivität der consist-Beziehung
17  consist(x,y) ∧ consist(z,y)
18    ⇒ (consist(x,z) ∨ consist(z,x)),
19
20  //Transitivität der require-Beziehung
21  require(x,y) ∧ require(y,z) ⇒ require(x,z),
22

```



```

23  consist(x,y) ∧ change(y,z) ⇒
24      ¬ read_only_use(x,z) ∧ change(x,z),
25
26  consist(x,y) ∧ create(y,z) ⇒
27      ¬ read_only_use(x,z) ∧ create(x,z),
28
29  consist(x,y) ∧ eliminate(y,z) ⇒
30      ¬ read_only_use(x,z) ∧ eliminate(x,z),
31
32  create(x,y) ∧ create(z,y) ⇒ consist(x,z) ∨ consist(z,x),
33
34  support(c,bp) ∧
35      (read_only_use(bp,bo) ∨ create(bp,bo) ∨
36         change(bp,bo) ∨ eliminate(bp,bo)) ⇒
37      ∃ s (depict(s,bo) ∧
38         (read_only_use(c,s) ∨ create(c,s)
39          ∨ change(c,s) ∨ eliminate(c,s))),
40
41  create(x,s) ∧ (read_only_use(y,s) ∨ change(y,s)
42     ∨ eliminate(y,s)) ∧ x ≠ y ⇒ require(y,x),
43
44  create(x,s1) ∧ create(y,s2) ∧ x ≠ y ∧
45     (consist(s1,s2) ∨ reference(s1,s2)) ⇒ require(x,y)
46  }

```

Listing 6.6: Spezifikation Komponente

In Listing 6.6 in Zeile 32 legen wir fest, dass eine Struktur nur von einer Komponente erstellt werden darf. Die Beziehung zwischen Komponente und Struktur darf nur dann mehrmals eine Erstellende sein, wenn es sich um eine Komponentenhierarchie handelt. Dies soll dazu führen, dass die Verantwortung über eine Struktur in einer Komponente bereits sehr früh im Softwareentwicklungsprozess festgelegt werden kann.

In Zeile 34 legen wir fest, dass wenn ein Geschäftsprozess ein Geschäftsobjekt involviert und von einer Komponente unterstützt wird, dieses Geschäftsobjekt softwareseitig abgebildet sein muss und dieses von der unterstützenden Komponente auch verwendet werden muss.

In Zeile 41 legen wir fest, dass Komponenten sich benötigen, sofern sie Strukturen verwenden, die von anderen Komponenten erstellt wurden.

In Zeile 44 legen wir fest, dass sich Komponenten benötigen, die Strukturen erstellen, die von der anderen Struktur entweder referenziert werden oder enthalten sind.

In Listing 6.7 wird der abstrakte Datentyp Struktur (*Structure*) spezifiziert.

```

1  SPEC STRUCTURE = {
2  sorts Structure , BusinessObject , BusinessObjectState ,
3      Maschine , OrganizationalUnit , Person|Role , Bool
4  operations
5  depict:      Structure , Maschine          → Bool ,

```

```

6   depict:   Structure , Person|Role       → Bool ,
7   depict:   Structure , OrganizationalUnit → Bool ,
8   depict:   Structure , BusinessObject    → Bool ,
9   depict:   Structure , BusinessObjectState → Bool ,
10  reference: Structure , Structure         → Bool ,
11  consist:  Structure , Structure         → Bool
12 axioms
13   //Partielle Ordnung, siehe Listing 6.2
14   consist ist partielle Ordnung,
15
16   //Exklusivität der consist-Beziehung
17   consist(x,y) ∧ consist(z,y)
18   ⇒ (consist(x,z) ∨ consist(z,x))
19 }

```

Listing 6.7: Spezifikation Struktur

6.3.3 Akteure

In diesem Abschnitt werden die Akteure des Modellierungsansatzes spezifiziert. In Listing 6.8 wird der abstrakte Datentyp Person bzw. Rolle (*Person|Role*), in Listing 6.9 die Organisationseinheit (*OrganizationalUnit*) und in Listing 6.10 die Maschine spezifiziert.

```

1 SPEC PERSON|ROLE = {
2 sorts Person|Role , BusinessProcess , Bool
3 operations
4   collaborate: Person|Role , BusinessProcess → Bool ,
5   responsible: Person|Role , BusinessProcess → Bool
6 axioms
7   responsible(x,y) ∧ responsible(z,y) ⇒ x = z
8 }

```

Listing 6.8: Spezifikation Person bzw. Rolle

```

1 SPEC ORGANIZATIONALUNIT = {
2 sorts Maschine , Person|Role , BusinessProcess , Bool
3 operations
4   consist:      OrganizationalUnit , OrganizationalUnit → Bool ,
5   has:          OrganizationalUnit , Maschine           → Bool ,
6   has:          OrganizationalUnit , Person|Role        → Bool ,
7   collaborate:  OrganizationalUnit , BusinessProcess    → Bool ,
8   responsible:  OrganizationalUnit , BusinessProcess    → Bool
9 axioms
10  //Partielle Ordnung, siehe Listing 6.2
11  consist ist partielle Ordnung,
12
13  responsible(x,y) ∧ responsible(z,y) ⇒ x = z
14 }

```

Listing 6.9: Spezifikation Organisationseinheit

```

1 SPEC MASCHINE = {
2   sorts Maschine , BusinessProcess , Bool
3   operations
4     consist:      Maschine , Maschine      → Bool ,
5     collaborate: Maschine , BusinessProcess → Bool ,
6     responsible: Maschine , BusinessProcess → Bool
7   axioms
8     //Partielle Ordnung, siehe Listing 6.2
9     consist ist partielle Ordnung ,
10
11    responsible(x,y) ∧ responsible(z,y) ⇒ x = z
12 }

```

Listing 6.10: Spezifikation Maschine

In Listing 6.8, 6.9 und 6.10 legen wir in den Zeilen 7, 13 und 11 fest, dass die Verantwortung von Geschäftsprozessen nur von einer Person bzw. Rolle, Organisationseinheit oder Maschine verantwortet werden kann.

6.3.4 Qualität und Bedingungen

Im Modellierungsansatz werden Qualitätseigenschaften und Bedingungen im Basismodell erfasst. Wir gehen in dieser Phase des Requirements Engineerings nicht über die Benennung der Qualitätseigenschaften und Bedingungen hinaus.

Listing 6.11 zeigt die Spezifikation des Konzeptes Qualität und Listing 6.12 zeigt die Spezifikation des Konzeptes Bedingung.

```

1 SPEC QUALITY = {
2   sorts Quality , BusinessProcess ,
3     BusinessObject , Structure , Component , Bool
4   operations
5     fulfill: BusinessProcess , Quality → Bool ,
6     fulfill: BusinessObject , Quality → Bool ,
7     fulfill: Structure , Quality → Bool ,
8     fulfill: Component , Quality → Bool

```

Listing 6.11: Spezifikation Qualitätseigenschaft

```

1 SPEC CONSTRAINT =
2   sorts Constraint , BusinessProcess ,
3     BusinessObject , Structure , Component , Bool
4   operations
5     fulfill: BusinessProcess , Constraint → Bool ,
6     fulfill: BusinessObject , Constraint → Bool ,
7     fulfill: Structure , Constraint → Bool ,
8     fulfill: Component , Constraint → Bool
9 }

```

Listing 6.12: Spezifikation Bedingung

Auf Basis der in Artefakten erhobenen Inhalte haben wir festgestellt, dass keine Qualitätseigenschaften bzw. Bedingungen für Akteure erfasst wurden. Wir haben daher diese nicht in die Spezifikation aufgenommen.

6.4 Qualitätseigenschaften eines Basismodells

Mit dem vorgestellten Ansatz werden Basismodelle erstellt, die konform zur ontologischen Basis sind. Für Ontologien wurden bereits objektive und subjektive Kriterien zur Feststellung der Qualität erarbeitet, siehe dazu Abschnitt 2.3.8.

In diesem Abschnitt werden diese erarbeiteten Qualitätskriterien mit bekannten Qualitätskriterien der IEEE für Anforderungen [IEE98] betrachtet. Die Qualitätskriterien an Anforderungen der IEEE sehen wir als relevant an, da das Basismodell als Produkt des Ansatzes ein Artefakt des Requirements Engineering ist und dadurch auch dieselben Qualitätskriterien zu erfüllen hat.

6.4.1 Abgestimmt

Abgestimmt ist ein Basismodell dann, wenn es von allen relevanten Stakeholder akzeptiert wurde.

Für Anforderungen gilt, dass sie abgestimmt sind, wenn sie für alle Stakeholder korrekt sind und für alle Stakeholder als gültige Anforderungen akzeptierbar sind [RS09]. Ein Basismodell ist durch den kooperativen Erstellungsvorgang und den involvierten Stakeholdern per se inhaltlich abgestimmt.

Wir adressieren die Problematik, diesen Abstimmungsbedarf über die in die Erstellung involvierten Stakeholder hinaus zu minimieren, indem die an der Erstellung beteiligten Rollen so gewählt wurden, dass sie ihre (fachliche bzw. technische) Seite vertreten können.

6.4.2 Angemessen

Ein Basismodell ist angemessen, wenn das Gesamtverständnis⁵ so repräsentiert ist, dass es für den Zweck angemessen ist.

Der Zweck des Basismodell ist es, (a) inhaltliche Missverständnisse zu reduzieren, (b) als Basis für andere Artefakte dafür zu sorgen, dass sich diese Artefakte auf einer ähnlichen Abstraktionsebene befinden, (c) die Konsistenz der darauf basierenden Artefakte untereinander zu verbessern, (d) die Modellbildung so zu lenken, dass ähnliche Sachverhalte auch ähnlich repräsentiert werden, (e) eine ausgeglichene Repräsentation von Wissen in allen auf dem Basismodell basierenden Artefakten zu erreichen, (f) die Möglichkeit der Verknüpfung zu anderen Artefakt-Typen zu bieten und (g) ein stakeholderweites Verständnis zu repräsentieren.

⁵ In Kontext dieser Arbeit ist das Gesamtverständnis das gemeinsame Verständnis aller relevanten Stakeholder über die Problem- und Lösungsdomäne auf einer gemeinsam festgelegten Abstraktionsebene. Die Definition befindet sich auf Seite 4.

Die Feststellung, ob das Basismodell für den Zweck angemessen ist, wird im Rahmen der Evaluierung getroffen.

6.4.3 Eindeutig

Eindeutig ist ein Basismodell dann, wenn alle an der Erstellung beteiligten Stakeholder einzeln nach der Bedeutung von erfassten Entitäten und deren Beziehungen befragt, den gleichen Inhalt eines Basismodells in ihren Worten wieder geben können. Wir beziehen daher die Eindeutigkeit auf die Personen, die an der Erstellung beteiligt waren.

Die Eindeutigkeit muss darüber hinaus, zumindest auch für die Personen gegeben sein, die das Basismodell in weiteren Phasen des Softwareentwicklungsprojektes verwenden, um z. B. auf Basis des Basismodells Artefakte in anderen Modellierungstechniken zu erstellen.

Die Konzepte, die im Basismodell verwendet werden um die Entitäten zu stereotypisieren, sind disjunkt. Dies gilt auch für die stereotypisierten Beziehungen. Wir haben versucht, die Eindeutigkeit zumindest auf einer konzeptuellen Ebene zu gewährleisten. Die Entitäten sind dadurch zumindest in ihrer konzeptuellen Bedeutung eindeutig.

6.4.4 Korrekt

Ein Basismodell ist korrekt, wenn es die inhaltlichen Vorstellungen, der in der Erstellung involvierten Stakeholder, richtig wiedergibt.

Im Vorgehen zur Modellbildung im Modellierungsansatz, siehe dazu Kapitel 7, ist die Verifikation der erhobenen Begriffe und deren Beziehungen, die später als Entitäten und Beziehungen im Basismodell erfasst sein können, Teil des Modellierungsansatzes. Wir stellen daher nach jeder Erweiterung des Basismodells sicher, dass es die Vorstellungen der Stakeholder noch korrekt wieder gibt.

6.4.5 Konsistent

Konsistent ist ein Basismodell dann, wenn es keine inhaltlichen Widersprüche im Basismodell gibt.

Es ist nicht möglich, alle inhaltlichen Widersprüche anhand der gewählten Repräsentation zu finden. Wir haben im Rahmen der Entwicklung der ontologische Basis Axiome erarbeitet. Diese Axiome sollen zumindest sicherstellen, dass die inhaltlichen Widersprüche, die aufgrund der Konzepte und deren Beziehungen gefunden werden können, nicht auftreten.

Die gewählte Repräsentation muss einfach verwendbar und erstellbar sein und dennoch, für frühe Phasen des Requirements Engineerings, noch genügend Aussagekraft haben, um eine angemessene Repräsentation der Problem- und Lösungsdomäne zu gewährleisten.

Konsistenz von Artefakten verknüpft mit einem Basismodell. Zusätzlich zur Konsistenz eines Basismodells sprechen wir in dieser Arbeit auch davon, dass wir Inkonsistenzen zwischen Artefakten, die mit einem Basismodell verknüpft sind, reduzieren. Artefakte, die in frühen Phasen erstellt werden, werden häufig unabhängig voneinander erstellt. Widersprüche zwischen den Artefakten sind nur schwer zu identifizieren (siehe dazu Kapitel 3). Im Modellierungsansatz verknüpfen wir diese Artefakte mit einem Basismodell und stellen sicher, dass die Artefakte sich nicht mit dem Basismodell widersprechen. Wir sprechen von einem Widerspruch zwischen einem Basismodell und einem damit verknüpften Artefakt dann, wenn Aussagen im Basismodell, formuliert mit Hilfe von Entitäten und deren stereotypisierten Beziehungen untereinander, die in Ausschnitten oder im Ganzen in einem verknüpften Artefakt detailliert wurden, widersprüchlich zu den Aussagen im Basismodell sind.

6.4.6 Modifizierbar

Die Modifizierbarkeit von Basismodellen ist wesentliches Kriterium der Modellierung. Modellierung ist eine iterative und inkrementelle Aktivität, Änderungen sind daher allgegenwärtig, wenn z. B. durch Detaillierung festgestellt wird, dass sich Beziehungen zwischen Begriffen der Domäne verändern.

Beispiel: In einer Detaillierung wird ein Geschäftsprozess (p) in zwei weitere unterteilt (p_1 und p_2), daher $\text{consist}(p, p_1) \wedge \text{consist}(p, p_2)$. p involviert ein Geschäftsobjekt (g) rein lesend (read-only-use). In der Detaillierung wird festgestellt, dass der Geschäftsprozess p_1 das Geschäftsobjekt g nun ändernd (change) involviert. Dies hat Auswirkung auf p , da die Aussage, dass das Geschäftsobjekt g nur lesend involviert ist, nicht mehr richtig ist.

Wir sagen, dass ein Basismodell dann modifizierbar ist, wenn künftige Änderungen am Basismodell möglich sind, ohne dass die Konsistenz des Basismodells gefährdet ist. Wir stellen dies mit den Axiomen sicher. Ein Basismodell muss nach jeder Iteration konsistent sein.

6.4.7 Verständlich

Amann und Fleischmann [AF06] stellten fest, dass die Verständlichkeit eines Modells nicht nur vom betrachteten Modell, sondern auch vom Modellbetrachter abhängt.

Amann und Fleischmann [AF06] stellen weiter fest, dass einem Modellnutzer nur dann ein Modell von deren Bedeutung her klar sein kann, wenn der Modellnutzer die verwendeten Klassen der Entitäten kennt. Wir haben in dieser Arbeit sämtliche Klassen informal wie formal spezifiziert, um dies zu gewährleisten. Wir haben zusätzlich die möglichen Beziehungen zwischen diesen Klassen, die wir Konzepte nennen, spezifiziert.

Die im Basismodell modellierten Entitäten werden zusätzlich zur Stereotypisierung durch die Konzepte noch benannt. Wir schränken hier den Interpretationsraum noch weiter ein. Zusätzlich sind die Beziehungen zwischen diesen Entitäten stereotypisiert.

Ein Modell ist umso verständlicher, je weniger qualitative und quantitative Kom-

plexität es aufweist [AF06]. Die qualitative Komplexität bezeichnet die Komplexität durch Kompliziertheit und die quantitative Komplexität die Komplexität durch die Menge an Entitäten [AF06]. Wir haben in dieser Arbeit darauf geachtet, dass nur die Konzepte und Beziehungen zwischen diesen in der ontologische Basis vorkommen, die von den Modellerstellern in der Vergangenheit bereits verwendet wurden.

6.4.8 Vollständig

Ein Basismodell ist vollständig, wenn:

- (a) Alle Geschäftsprozesse, die vom zu entwickelnden Softwaresystem unterstützt werden sollen, im Modell erfasst wurden,
- (b) alle Geschäftsobjekte, die in diesen Geschäftsprozessen involviert sind, erfasst wurden,
- (c) alle Komponenten erfasst wurden, die diese Geschäftsprozesse unterstützten sollen und
- (d) alle Akteure erfasst wurden, die in die in (a) erwähnten Geschäftsprozesse, in irgendeiner Form, involviert sind und
- (e) alle Geschäftsobjekte und Akteure softwareseitig erfasst wurden, in Form von Strukturen, die in diesen Komponenten in irgendeiner Form verwendet werden und
- (f) alle Qualitätseigenschaften und Bedingungen benannt wurden, die es in der Entwicklung zu berücksichtigen gilt.

Die Vollständigkeit von Anforderungen, wie auch jene des Basismodells ist schwer festzustellen. Wir haben versucht die Kompetenzfragen⁶ entsprechend der in der Aufzählung aufgeführten Kriterien zu gestalten. Werden diese Fragen im Rahmen der Erhebung beantwortet, ist das Basismodell vollständig.

6.5 Zusammenfassung

Dieses Kapitel stellte einführend das Beispiel eines Basismodells dar. In Abschnitt 6.2 wurde das erfasste Wissen informal beschrieben. Formal wurde dies in Abschnitt 6.3 gemacht. Das Kapitel schließt mit der Beschreibung der Qualitätseigenschaften eines Basismodells in Abschnitt 6.4 ab.

⁶ Eine Kompetenzfrage ist im Rahmen der Ontologieentwicklung eine Frage in natürlicher Sprache, die dabei hilft (a) das Scoping einer Ontologie festzulegen und (b) die Ontologie zu evaluieren [AGP04]. Eine Kompetenzfrage muss durch die entwickelte Ontologie beantwortbar sein. Die Definition befindet sich auf Seite 148.

Konstruktionsprozess und Verknüpfung zu Artefakten

Die Erstellung eines Basismodells ist eine Kernaufgabe des Modellierungsansatzes dieser Arbeit. Dieses Kapitel beschreibt die involvierten Rollen, den Konstruktionsprozess eines Basismodells, sowie beispielhaft, die Vorgehensweise zur Verknüpfung zu anderen Artefakt-Typen.

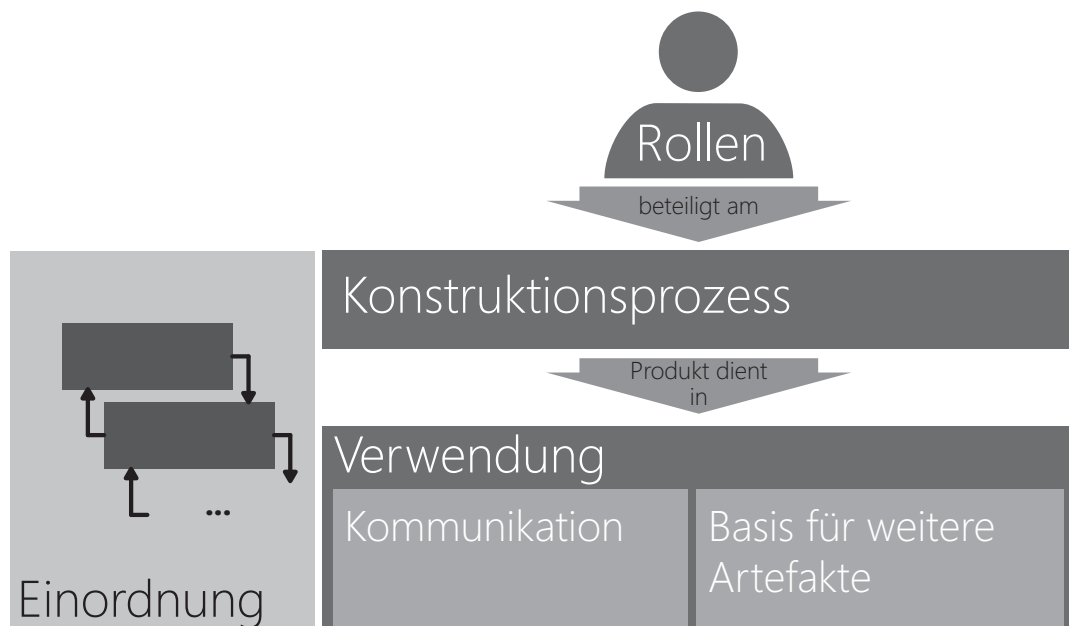


Abbildung 7.1: Aufbau des Kapitels

Abbildung 7.1 zeigt den Aufbau des Kapitels. Das Kapitel gliedert sich in folgende Abschnitte:

Einordnung. In Abschnitt 7.1 nehmen wir eine Einordnung des Konstruktionsprozesses und der Verwendung eines Basismodells im Softwareentwicklungsprozess und in die Artefaktorientierung vor.

Rollen. In Abschnitt 7.2 führen wir in die am Konstruktionsprozess beteiligten Rol-

len mit ihren Aufgaben und Verantwortungen ein.

Konstruktionsprozess. In Abschnitt 7.3 stellen wir den Konstruktionsprozess vor, dessen Produkt ein Basismodell ist. Im Abschnitt 7.3 sprechen wir auch über die Verwendung des Basismodells bezüglich Kommunikation zwischen den Stakeholdern, sowie über den Vorgang die Kommunikationslücke zwischen diesen zu verkleinern, indem eine gemeinsame Bedeutung der zu modellierenden Sachverhalte identifiziert wird.

Verknüpfung zu anderen Artefakten. In frühen Phasen des Requirements Engineerings werden eine Vielzahl von Artefakten erstellt. In Abschnitt 7.4 zeigen wir beispielhaft, wie eine Verknüpfung des im Rahmen des Konstruktionsprozesses erstellten Basismodells zu anderen Artefakten operationalisiert werden kann.

7.1 Einordnung des Modellierungsansatzes

Dieser Abschnitt beschreibt die Einordnung des Modellierungsansatzes in den Softwareentwicklungsprozess (in Abschnitt 7.1.1) und in die Artefaktorientierung (in Abschnitt 7.1.2).

7.1.1 Einordnung in den Softwareentwicklungsprozess

Ein typischer Softwareentwicklungsprozess besteht zumindest aus den Phasen Requirements Engineering, Entwurf, Implementierung und Testen, dies wird auf der linken Seite der Abbildung 7.2 dargestellt. Der hier vorgestellte Ansatz wird im Requirements Engineering angewendet. Requirements Engineering besteht typischerweise aus den Phasen Erhebung, Analyse/Modellierung, Spezifikation und Validierung [AW05], dies wird auf der rechten Seite der Abbildung 7.2 dargestellt.

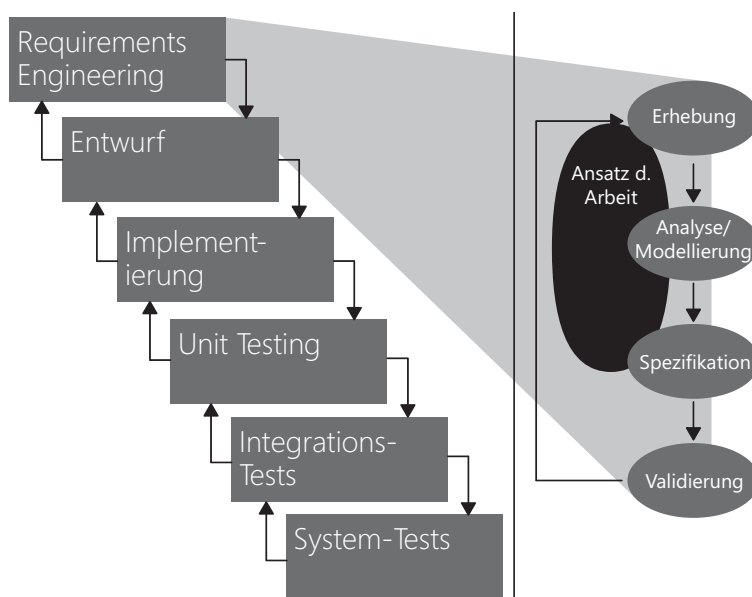


Abbildung 7.2: Einordnung in den Softwareentwicklungsprozess und dem Requirements Engineering

- ▶ **Erhebung:** Erhebung von Anforderungen durch die Kommunikation mit den relevanten Stakeholdern des Auftraggebers (z. B. Benutzer, Domänenexperten).
Beitrag der Arbeit: Die Erhebung einer gemeinsamen Vorstellung des Problems, sowie Anforderungen an eine mögliche Lösung, wird durch einen vorgegeben Erhebungsprozess zwischen den verschiedenen involvierten Stakeholdern gesteuert. Das Produkt dieses Prozesses ist ein Modell (das Basismodell), das diese Vorstellung über Problem und Anforderungen an die Lösung abbildet. Dieses Basismodell wird in den weiteren Phasen des Requirements Engineering an verschiedenen Stellen eingesetzt.
- ▶ **Analyse/Modellierung:** Ausarbeitung von Modellen, die unterschiedliche Sichtweisen auf die Problemdomäne repräsentieren. In der Analyse wird festgelegt, welche Ausschnitte der modellierten Sichten Ausschnitte der Umgebung des zu entwickelnden Systems sein sollen.
Beitrag der Arbeit: Die Verknüpfung des im Rahmen der Anwendung des Modellierungsansatzes erstellten Basismodells zu anderen Artefakten, wird im Rahmen der Arbeit beispielhaft für wichtige Artefakt-Typen des Requirements Engineerings gezeigt.
- ▶ **Spezifikation:** Einzelne Modelle bzw. Modellelemente werden in diesem Schritt präzise beschrieben.
Beitrag der Arbeit: Auf Basis des Basismodells entstehen durch die Verknüpfung zu anderen Artefakt-Typen besser strukturierte Artefakte. Der Zusammenhang zwischen den Artefakten wird durch das durch den Modellierungsansatz erstellte Basismodell expliziert.
- ▶ **Validierung:** Kontrolle der Korrektheit, Vollständigkeit und Konsistenz. Es wird die Frage beantwortet, ob das *richtige* System gebaut wird?

Verknüpfung zu lösungsspezifischen Artefakten. Partsch [Par10] stellt fest, dass es häufig keine klare Abgrenzung zwischen den Phasen Entwurf und Requirements Engineering gibt. Er führt weiter aus, dass im Requirements Engineering oft auch eindeutige Vorgaben oder Lösungskonzepte für den Entwurf und manchmal sogar auch für die Implementierung festgeschrieben werden.

Wir trennen diese zwei Welten im Ansatz. Wir zeigen jedoch beispielhaft, wie Verknüpfungen zu Artefakten, die lösungsspezifisch sind (z. B. Komponentendiagramme), aussehen könnten.

7.1.2 Einordnung in die Artefaktororientierung

Die Idee hinter der Artefaktororientierung ist es, sich bei der Entwicklung eines Softwaresystems auf die zu erstellenden Artefakte zu konzentrieren und nicht auf die anzuwendenden Methoden oder Prozessschritte [FLPW11].

Im Rahmen einer artefaktbasierten Vorgehensweise wird ein Referenzmodell erstellt, welches alle Artefakte enthält, die im Laufe des Entwicklungsprojektes zu erstellen sind. Hierzu werden die Artefakte, die Beschreibungsmittel und die Struktur bestimmt. Im Projekt wird ein Prozess definiert, indem man sich einigt, welche Rollen, welche Artefakte erstellen und festlegt, bis zu welchen Meilenstein dies geschehen soll.

Méndez Fernández et al. [FLPW11] haben im Rahmen eines Industrieprojektes ein Referenzmodell erstellt, das sie BISA (*business information systems analysis*) nennen. Abbildung 7.3 zeigt dieses Referenzmodell.

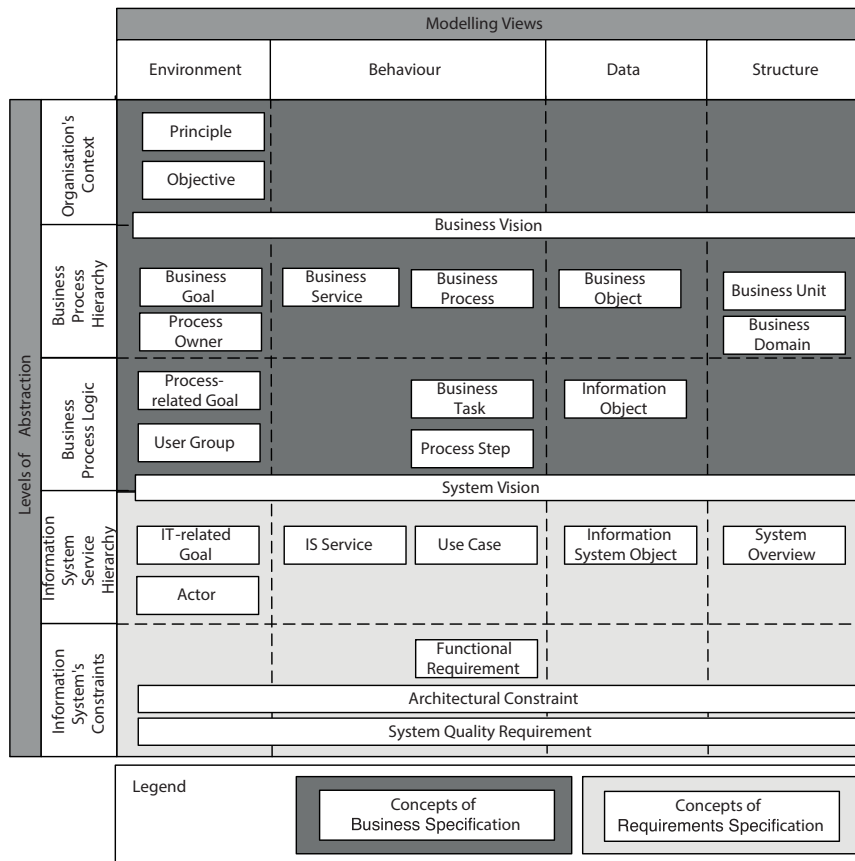


Abbildung 7.3: Artefakt-Modell: Das BISA-Referenzmodell [FLPW11]

Das in Abbildung 7.3 vorkommende Artefakt *System Vision* dient u.a. dem Scoping¹. Nur ein Teil des in Artefakten explizierten Wissens wird in den nächsten Abstraktionsebenen benötigt [Fer11]. Bevor detailliert Anforderungen erhoben werden, wird der Anwendungsbereich des Systems mit dem Artefakt *System Vision* definiert. Die *System Vision* muss mit allen relevanten Stakeholdern abgestimmt sein [Fer11].

Diese Arbeit schlägt eine Ausprägung des Artefaktes *System Vision* vor. Wir geben vor, (a) wie die Struktur dieses Artefaktes aufgebaut ist, (b) welche Inhalte erfasst werden, (c) wie die Erhebung des Artefaktes vorgenommen werden soll, (d) wie die Konsistenz des Artefaktes sichergestellt werden kann und (e) wie eine mögliche Verknüpfung zu anderen Artefakt-Typen (z. B. Anwendungsfällen) konkret vorgenommen werden kann.

¹ Scoping leitet sich aus dem Englischen (*scope*) ab und bedeutet soviel wie die Festlegung eines Geltungsbereiches bzw. das Abgrenzen eines Bereiches.

7.2 Rollen

An der Erstellung des Basismodells sind immer mehrere Rollen beteiligt. Ein Basismodell enthält konsensuelles Wissen. Es ist das Produkt einer Tätigkeit, an der mehrere Personen beteiligt sind. Der Modellierungsansatz definiert die in Abbildung 7.4 abgebildeten Rollen. Diese Rollen sind direkt an der Erstellung des initialen Basismodells beteiligt. Die einzige Rolle, die operativ am Basismodell Änderungen vornimmt, ist die Rolle des Modellierers, alle anderen Rollen liefern Diskussionsbeiträge bzw. bringen ihre Expertise ein.



Abbildung 7.4: Die involvierten Rollen im Modellierungsansatz, der Modellierer ist die Rolle, die Änderungen am Basismodell durchführt

Zur Auswahl der Rollen. Prieto-Diaz führt zur Domänenanalyse folgende Rollen ein [PD88]:

- ▶ **Domänen-Analyst:** Der Analyst bringt den gewählten Modellierungsansatz in die Analyse mit ein, u.a. in Form von Richtlinien. Er verantwortet die erstellten Artefakte (z. B. Taxonomien). Im Modellierungsansatz dieser Arbeit, wird diese Rolle Modellierer genannt. Wir begründen diese Entscheidung damit, dass der Modellierer zwischen der typischerweise Domäne genannten Problem- bzw. Anwendungsdomäne und den Softwareentwicklern steht und im Rahmen dieser Arbeit keiner dieser Seiten zugeordnet werden soll.
- ▶ **Domänenexperte:** Der Domänenexperte bringt sein Wissen über eine Domäne ein. Dieses Wissen ist nicht leicht zu extrahieren, da auch von Domänenexperten nicht alle Nuancen von Problemen in Betracht gezogen werden [CY94].
- ▶ **Softwareentwickler:** Verwendet das explizierte Wissen. Die Rolle wird im Modellierungsansatz übernommen. Softwareentwickler spielen in der Literatur der Domänenanalyse eine untergeordnete Rolle. Softwaresysteme wirken auf die Domäne zurück (z. B. in Form von Legacy-Systemen oder in Form des zu erstellenden Softwaresystems), siehe dazu auch [Ste93]. Diese Interaktionspunkte dienen uns als Anknüpfungspunkte um in frühen Phasen des Requirements Engineerings die zu unterstützenden Geschäftsprozesse zu identifizieren. Diese unterstützende Rolle des Softwaresystems muss den Domänenexperten wie auch den Softwareentwicklern klar sein um eine gemeinsame Vorstellung zu finden. Wir sehen daher auch Softwareentwickler als wichtige Rolle an. Wobei die Aufgaben der Softwareentwickler über die, die in der Domänenanalyse genannt werden (z. B. in [PD88]), hinaus gehen.

Zur Anzahl der Domänenexperten und Softwareentwickler. Hier sei auf die Grundlagen *gemeinsame Modellierung* in Abschnitt 2.2.4 verwiesen. Ausgehend davon, dass die Modellierung, z. B. eines Geschäftsbereiches, durch einen Experten häufig von einer persönlichen Vorstellung des Geschäftsbereiches getrieben ist, kann es vorkommen, dass gerade jener Fachbereich, der einen Domänenexperten besonders interessiert, überrepräsentiert ist. Williams [Wil00] stellte in ihrer Arbeit fest, dass zwei Personen einer Fachdomäne sich bei der Diskussion über einen Sachverhalt bereichern und zu besseren Lösungen kommen. Gründe dafür führt Flor et al. [FH91] auf, u.a. liegt das an eher unterschiedlichen Sichtweisen von unterschiedlichen Personen auf ein Problem.

Dieser Abschnitt führt die im Modellierungsansatz beteiligten Rollen mit ihren Aufgaben und Verantwortungen ein.

7.2.1 Modellierer

Der Modellierer ist jene Rolle, die operative Änderungen am Basismodell durchführt. Der Modellierer stammt typischerweise nicht aus der Gruppe der Domänenexperten, sondern wird wahrgenommen durch einen Anforderungsingenieur oder einen Softwareentwickler. Diese Entscheidung ist dem Sachverhalt geschuldet, dass Anforderungsingenieure bzw. Softwareentwickler darauf geschult sind, Detailfragen zum Geschäftsbereichswissen zu stellen, um diese angebracht im Basismodell abbilden zu können.

Aufgabe. Aufgabe des Modellierers ist es, operative Arbeiten am Basismodell durchzuführen. Dies ist u.a. im ersten Schritt, das Erheben der Begrifflichkeiten anhand der Vorgehensweise (siehe Abschnitt 7.3). Darüberhinaus werden Änderungen am Basismodell nur durch den Modellierer durchgeführt. Dieser koordiniert auch die Diskussion rund um das Basismodell.

Verantwortung. Der Modellierer verantwortet die Konsistenz und Qualität des Basismodells und leitet die Diskussion.

7.2.2 Softwareentwickler bzw. Anforderungsingenieur

Der Softwareentwickler bzw. Anforderungsingenieur ist die Rolle, die primär für das Wissen in der Lösungsdomäne und der Klärung von Abhängigkeiten zum Wissen der Problemdomäne zuständig ist.

Aufgabe. Die Aufgabe der Softwareentwickler bzw. Anforderungsingenieure ist es, einerseits die Vorstellung über Abhängigkeiten im Wissen über die Lösungsdomäne untereinander zu festigen und andererseits die Anknüpfungspunkte zu Geschäftsprozessen und Geschäftsobjekten (z. B. wie werden Geschäftsprozesse unterstützt) zu präzisieren.

Verantwortung. In der Verantwortung des Softwareentwicklers bzw. des Anforderungsingenieurs liegt es, eine konsensuelle Vorstellung über das Wissen der Lösungsdomäne untereinander zu festigen. Dieses Übereinkommen ist für die Folgearbeiten entscheidend, da nach diesem konsensuellen Übereinkommen und mit der Feststellung der involvierten Geschäftsprozesse mit der Aufteilung der Arbeit an die Softwareentwickler des Entwicklungsteams begonnen werden kann.

7.2.3 Domänenexperten

Der Domänenexperte ist die Rolle, die Wissen über die Problemdomäne im Detail einbringt.

Aufgabe. Die Aufgabe der Domänenexperten ist es, ihr Fachwissen über die Problemdomäne zu explizieren. Besondere Beachtung finden dabei die Ausschnitte der Problemdomäne, die für das zu entwickelnde Softwaresystem relevant sind.

Verantwortung. Es liegt in der Verantwortung der Domänenexperten eine gemeinsame Vorstellung über die Problemdomäne zu erarbeiten. Dieses konsensuelle und im Laufe der Modellierung explizierte Wissen ist integraler Bestandteil des Basismodells und es liegt daher in der Verantwortung der Domänenexperten, sich über diesen Bestandteil einig zu werden.

Es liegt in der Verantwortung der Domänenexperten die Vorstellungen des Benutzers über die Problemdomäne, in den der Benutzer interagiert bzw. Teile verantwortet, ebenso zu berücksichtigen.

7.2.4 Zusammenstellung der Rollen

Die Personen, die an der Erstellung eines Basismodells beteiligt sind, müssen in der Lage sein, über das gefundene konsensuelle Wissen zu kommunizieren. Dies involviert vor allem (a) Kompetenz in der Sache und (b) Kompetenz in der Weisung. Repräsentative Personen zu finden, die diese Eigenschaften erfüllen, ist daher von signifikanter Bedeutung für den Modellierungsansatz.

Häufig ist dies dann gegeben, wenn seitens der Auftragnehmer mindestens zwei Personen gefunden werden, die Erfahrung in der Softwareentwicklung haben und deren Einschätzungen bezüglich Aufwand und Machbarkeit anerkannt gut sind (daher von den anderen Softwareentwicklern des Auftragnehmers akzeptierbar sind).

Analog gilt die Aussage über die Softwareentwickler auch für die Domänenexperten. Mindestens zwei Domänenexperten sollten an der Erstellung eines Basismodells beteiligt sein. Dies ist notwendig, da die Domänenexperten ebenso eine gemeinsame Vorstellung über die Problemdomäne finden müssen.

7.3 Konstruktionsprozess

Dieser Abschnitt beschreibt den Konstruktionsprozess des Basismodells. Abbildung 7.5 zeigt einen Überblick über den Konstruktionsprozess.

Dieser besteht aus mehreren Schritten. (a) der Erhebung durch Kompetenzfragen. Im Rahmen der Beantwortung einer Kompetenzfrage werden Begriffe identifiziert. Diese Begriffe werden in ihrer Bedeutung Konzepten bzw. stereotypisierten Beziehungen zugeordnet oder aussortiert. Wenn die Begriffe Konzepten bzw. Beziehungen zugeordnet werden, nennen wir sie Entitäten bzw. stereotypisierte Beziehungen. Anschließend werden Beziehungen zu anderen Entitäten (inklusive denen, die bereits im Modell erfasst sind) identifiziert. Die identifizierten Entitäten und stereotypisierten Beziehungen werden im Rahmen der Modellierung in das Basismodell übernommen bzw. das Basismodell wird angepasst.

Definition **Kompetenzfrage:**

Eine Kompetenzfrage ist im Rahmen der Ontologieentwicklung eine Frage in natürlicher Sprache, die dabei hilft (a) das Scoping einer Ontologie festzulegen und (b) die Ontologie zu evaluieren [AGP04]. Eine Kompetenzfrage muss durch die entwickelte Ontologie beantwortbar sein.

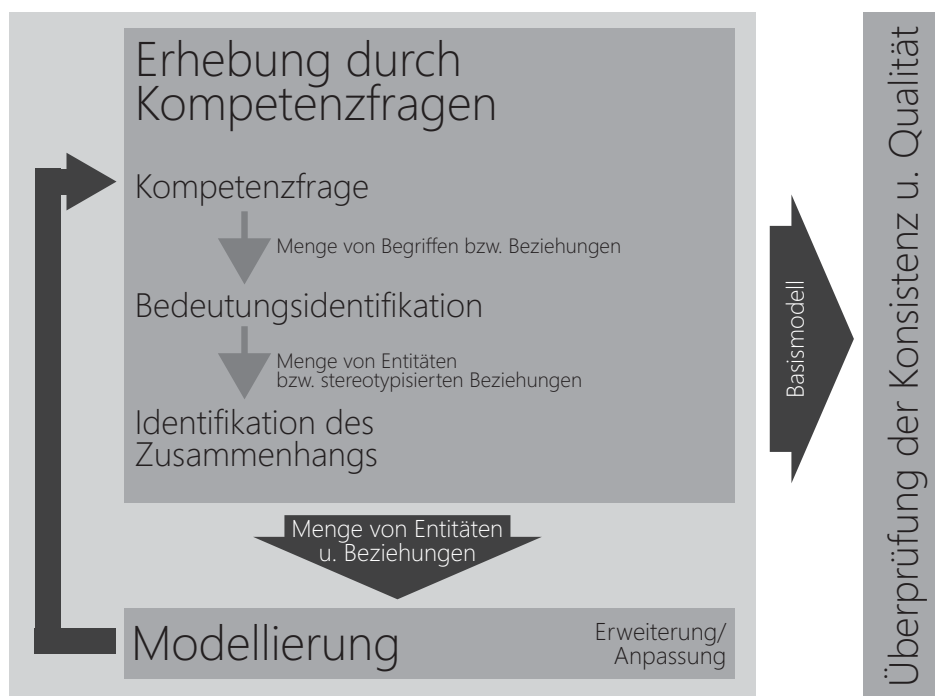


Abbildung 7.5: Überblick über eine Iteration des Konstruktionsprozesses

Schritte (a) und folglich auch (b) werden so lange wiederholt, bis alle Kompetenzfragen beantwortet wurden und das Abbruchkriterium erreicht wurde. Nach diesen Schritten, wird in Schritt (c) das Basismodell auf seine Konsistenz überprüft.

Nach der Prüfung der Konsistenz des Basismodells (siehe Abschnitt 7.3.3) ist eine Iteration abgeschlossen. Je nach gewünschtem Detaillierungsgrad (z. B. aufgrund der

Verknüpfung zu anderen Artefakten), werden diese Schritte wiederholt.

Zur Erarbeitung des Konstruktionsprozesses. Der Konstruktionsprozess basiert auf Ansätzen zur Ontologieerstellung (siehe dazu Abschnitt 2.3.5), auf Ansätzen zur Domänenanalyse (u.a. [PD90, PDA91, FV99, ZV07]), aus Erkenntnissen zur Modellbildung (siehe dazu Abschnitt 2.2) und aus Vorgehensweisen, erhoben aus der Praxis zum Finden eines initialen Verständnisses über die Problem- und Lösungsdomäne (siehe dazu Kapitel 3).

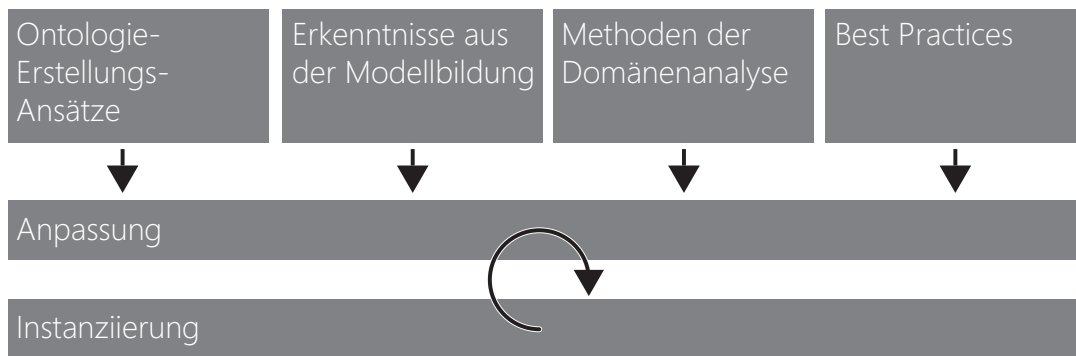


Abbildung 7.6: Erarbeitung der Vorgehensweise

Abbildung 7.6 zeigt die schrittweise Vorgehensweise, wie wir den Konstruktionsprozess erarbeitet haben. Die erwähnten Ansätze der Ontologieerstellung sowie der Domänenanalyse versuchen Wissen zu explizieren (wobei sie sich im Zweck und Vorgehen unterscheiden). Wir haben für das Lösen unserer Problemstellung Teile der Ontologieerstellung (z. B. Identifikation von relevantem Wissen) und aus der Domänenanalyse (Schrittweises Vorgehen und Rollen) verwendet, um den Ansatz zu erarbeiten.

Der Konstruktionsprozess wurde wiederholt angewendet (im Rahmen von Softwareprojekten in der Praxis und in Softwareprojekten im Rahmen von Praktika an der TU München) und weiter angepasst.

Zur Erarbeitung der Kompetenzfragen. In diesem Abschnitt explizieren wir die Vorgehensweise, die zur Reihenfolge der einzelnen Schritte der Erhebung durch Kompetenzfragen und den untergeordneten Kompetenzfragen (siehe nächster Abschnitt 7.3.1) geführt hat.

Die einem Schritt untergeordneten Kompetenzfragen helfen die Aufgabe, die in einem Schritt formuliert wurde, zu erfüllen (z. B. Identifikation von relevanten Geschäftsprozessen). Durch die Kompetenzfragen werden Begriffe identifiziert, die dann durch die Bedeutungsidentifikation zu Entitäten oder stereotypisierten Beziehungen werden. Die Kompetenzfragen wurden durch die in der Anwendung der Methode gewonnenen Erfahrungen wiederholend erweitert bzw. angepasst.

Die Reihenfolge der Schritte haben wir u.a. den Zusammenhängen der identifizierten Konzepten entnommen (siehe Abschnitt 6.2 und 6.3). Die Reihenfolge spiegelt daher, auf Geschäftsprozessen aufbauend, auch diese Zusammenhänge wieder. Aufbauend auf diesen Zusammenhang haben wir einen Dialog zwischen Softwareentwicklern

bzw. Anforderungsingenieuren und Domänenexperten formuliert. Nachfolgend explizieren wir diesen Zusammenhang zwischen den einzelnen Schritten mit Übergang zum nächsten Schritt:

Unternehmen haben typischerweise eine Vorstellung davon, welche Geschäftsprozesse für das Unternehmen relevant sind. Geschäftsprozesse spielen auch für die zu entwickelnde Software eine wesentliche Rolle, da Software im Rahmen von Geschäftsprozessen verwendet wird und innerhalb eines Geschäftsprozesses eine oder mehrere Funktionen erfüllen muss.

Siehe Schritt (a): *Identifikation von Geschäftsprozessen.*

Ausgehend von den Geschäftsprozessen identifizieren und benennen wir Teile (Komponenten) des geplanten Softwaresystems und setzen diese Komponenten mit den Geschäftsprozessen in Beziehung. Dieses Vorgehen ermöglicht es uns, die Komponenten der Software, die für das Unternehmen und später den Softwareentwicklern relevant sein werden, direkt zu benennen. Sowie die Geschäftsprozesse, die mit diesen Komponenten unterstützt werden sollen, zu explizieren.

Das Erfassen der Geschäftsprozesse und Komponenten in einem Basismodell verbindet die Domänenexperten, die in unserer Zielgruppe von Geschäftsprozessen sprechen, und die Softwareentwickler.

Siehe Schritt (b): *Identifikation von Komponenten.*

In die Ausführung von Geschäftsprozessen sind typischerweise verschiedene *Objekte* involviert. Dies können immaterielle wie materielle Objekte sein. Diese Objekte, die wir Geschäftsobjekte nennen, werden häufig softwareseitig abgebildet. Wenn Domänenexperten Geschäftsobjekte beschreiben, beschreiben sie die Eigenschaften dieser Geschäftsobjekte. Nicht alle Eigenschaften sind relevant für das Softwaresystem. Wir modellieren die Geschäftsobjekte explizit, um zwischen der Beschreibung einer Abbildung eines Geschäftsobjektes sowie der Beschreibung des Geschäftsobjektes selbst unterscheiden zu können und diesen Unterschied für alle Stakeholder explizit zu machen.

Siehe Schritt (c): *Identifikation von Geschäftsobjekten.*

Wie im Vorabschnitt erwähnt, werden Geschäftsobjekte häufig in Softwaresystemen abgebildet. Eine Abbildung enthält typischerweise nicht alle Eigenschaften des Geschäftsobjektes. Wir nennen diese Abbildung Struktur. Hier ist es für die Softwareentwickler wichtig die Eigenschaften zu identifizieren, die für das zu entwickelnde Softwaresystem relevant sind.

Siehe Schritt (d): *Identifikation von Strukturen.*

Geschäftsobjekte werden im Rahmen von Geschäftsprozessen auf irgendeine Art verwendet. Diese Verwendung kann von einer rein Geschäftsobjekt-unveränderten Verwendung bis zu einer Erstellung eines Geschäftsobjektes gehen. Die Art der Verwendung ist für das Softwaresystem häufig von Bedeutung, z. B. wenn dies softwareseitig unterstützt werden soll.

Siehe Schritt (e): *Identifikation, wie Geschäftsobjekte in Geschäftsprozesse involviert sind.*

Die Verwendung von Geschäftsobjekten in Geschäftsprozessen wirkt sich auch auf das Softwaresystem aus. Dies kann vielfach geschehen, z. B. durch die Feststellung,

dass es softwareseitig Beziehungen zwischen Strukturen und Komponenten gibt, die es zwischen Geschäftsprozessen und Geschäftsobjekten nicht gibt. Z. B. müssen Strukturen auch an einer Stelle im Softwaresystem erzeugt werden, dies muss auf Seiten der Geschäftsprozesse für Geschäftsobjekte nicht gelten.

Siehe Schritt (f): *Identifikation der Beziehungen zwischen den Strukturen.*

Geschäftsobjekte weisen häufig Zustände auf, z. B. Bestellungen können u.a. den Zustand *ausgeliefert* aufweisen. Diese Zustände sind auch für das Softwaresystem von Bedeutung, da diese bei Bedarf softwareseitig abgebildet werden.

Siehe Schritt (g): *Identifikation von Zuständen von Geschäftsobjekten.* Siehe Schritt (h): *Identifikation, wie Strukturen diese Zustände abbilden.*

Im nächsten Schritt identifizieren wir, basierend auf den vorher identifizierten Zustände, alle Geschäftsprozesse, die für das geplante Softwaresystem von Bedeutung sein könnten. Hierzu identifizieren wir die Geschäftsprozesse, die diese Zustände ändern bzw. von einer Änderung betroffen sind.

Siehe Schritt (i): *Identifikation von Geschäftsprozessen, die diese Zustände ändern, bzw. von der Änderung betroffen sind.*

Aufbauend auf die identifizierten Geschäftsprozesse und den Zuständen aus den Vorschriften, identifizieren wir im nächsten Schritt die Auswirkungen auf die Strukturen und Komponenten.

Siehe Schritt (j): *Identifikation wie Strukturen in Komponenten involviert sind.*

Geschäftsprozesse werden von Akteuren verantwortet und ausgeführt. Für das zu entwickelnde Softwaresystem kann dies aus vielen Gründen wichtig sein, z. B. wegen einer angestrebten Trennung der Funktionen des Softwaresystems aufgrund unterschiedlicher Akteure.

Siehe Schritt (k): *Identifikation von Akteuren.*

Diese Akteure werden häufig in Softwareprojekten sowie auch im Softwaresystem abgebildet. Z. B. aufgrund von Zugriffskontrollen auf bestimmte Funktionen. Zusätzlich können diese hierarchisch organisiert sein.

Siehe Schritt (l): *Identifikation, ob die gefundenen Akteure in Strukturen abgebildet werden müssen.*

In Unternehmen – wie auch im zu entwickelndem Softwaresystem – gibt es zu erfüllende Qualitätseigenschaften und/oder zu erfüllende (Zwangs-)Bedingungen, die Auswirkung auf das zu entwickelnde Softwaresystem haben. Im Rahmen dieses Schrittes versuchen wir diese zu identifizieren.

Siehe Schritt (m): *Identifikation von Qualitätseigenschaften und Bedingungen.*

Wir erheben keinen Anspruch auf Vollständigkeit der von uns identifizierten Kompetenzfragen. Wir sehen die von uns im folgenden Abschnitt (7.3.1) identifizierten Kompetenzfragen als erste Aufstellung von Fragen, die die Erstellung eines Basismodells im Requirements Engineering leiten.

Die Kompetenzfragen können je nach Teildomäne noch angepasst bzw. erweitert werden. Z. B. ist es denkbar, dass Änderungen an Geschäftsobjekten noch weiters detailliert werden sollen (z. B. durch Fragen zur Zusammensetzung von einzelnen

Geschäftsobjekten).

Begrifflichkeiten. Abbildung 7.7 zeigt den Zusammenhang zwischen wichtigen Begriffen des Modellierungsansatzes. Begriffe werden mit Hilfe der Kompetenzfragen, siehe dazu Abschnitt 7.3, erhoben. Mit Hilfe der Kompetenzfragen wird bereits ein erstes Scoping² durchgeführt.

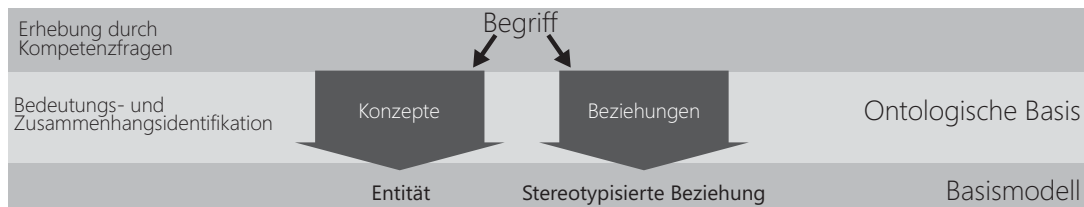


Abbildung 7.7: Zusammenhang der Begriffe

Mit Hilfe der Konzepte und Beziehungen in der ontologischen Basis werden die Bedeutung von Begriffen und mögliche Zusammenhänge mit anderen Begriffen (die wir dann Entitäten nennen) identifiziert. Wir sprechen dann von einer Entität, wenn ein Begriff sich die Eigenschaften genau eines Konzeptes teilt und wir sprechen dann von stereotypisierter Beziehung, wenn sich ein Begriff die Eigenschaften genau einer Beziehung aus der ontologischen Basis teilt, die als Stereotyp für diese stereotypisierte Beziehung dient. Ein Begriff ist entweder als Entität oder als stereotypisierte Beziehung in einem Basismodell repräsentiert.

7.3.1 Erhebung durch Kompetenzfragen

Die Identifikation der zu modellierenden Entitäten erfolgt durch eine Reihe von Kompetenzfragen. Die Kompetenzfragen sind einem übergeordneten Ziel untergeordnet. Z. B. ist das Ziel, das im Rahmen der Beantwortung der Kompetenzfragen in Schritt (b) erreicht werden soll, die Identifikation von Komponenten, die einen bestimmten Geschäftsprozess unterstützen sollen.

Die Beantwortung der Kompetenzfragen ist ein Dialog zwischen Domänenexperten (die wir mit R-DE abkürzen) und Softwareentwicklern bzw. Anforderungsingenieuren (die wir zusammenfassen und mit R-RE abkürzen).

Nachstehend explizieren wir den Dialog zwischen Domänenexperten und Softwareentwicklern bzw. Anforderungsingenieuren. Dieser besteht aus einer Reihe von Schritten. Innerhalb jedes Schrittes müssen eine Reihe von Kompetenzfragen beantwortet werden.

(a) Identifikation wichtiger (bzw. Detaillierung von) Geschäftsprozesse(n):

(a-i) Welche Geschäftsprozesse sollen zum Erreichen der Ziele unterstützt werden?

Falls nicht erste Iteration:

² Scoping leitet sich aus dem Englischen (*scope*) ab und bedeutet soviel wie die Festlegung eines Geltungsbereiches bzw. das Abgrenzen eines Bereiches.

(a-ii) Welche Geschäftsprozesse müssen für den gewünschten Detaillierungsgrad präzisiert werden?

(a-iii) Wie sind diese Geschäftsprozesse aufgebaut (aus welchen Einzelschritten bestehen diese)?

(a-iv) In welchen Beziehungen stehen die Geschäftsprozesse untereinander?

Verantwortliche Rolle für die Beantwortung: R-DE

(b) Identifikation von Komponenten, die diese Geschäftsprozesse unterstützen sollen:

(b-i) Welche Komponenten sollen diese Geschäftsprozesse aus Schritt (a) bzw. aus Schritt (i) unterstützen?

Verantwortliche Rolle für die Beantwortung: R-RE

(c) Identifikation von involvierten Geschäftsobjekten:

(c-i) Welche Geschäftsobjekte sind in den Geschäftsprozessen, identifiziert in Schritt (b), involviert?

(c-ii) Bestehen die erhobenen Geschäftsobjekte aus anderen nennenswerten Geschäftsobjekten (d.h. sie bestehen aus anderen Teil-Geschäftsobjekten)?

Verantwortliche Rolle für die Beantwortung: R-DE

(d) Identifikation von softwareseitigen Strukturen, die diese Geschäftsobjekte abbilden sollen:

(d-i) Wie werden die Geschäftsobjekte aus Schritt (c) softwareseitig abgebildet?

(d-ii) Wie stehen diese erhobenen Strukturen in Beziehung zueinander (referenzierend, komponierend)?

Verantwortliche Rolle für die Beantwortung: R-RE

(e) Identifikation, wie diese Geschäftsobjekte in den Geschäftsprozessen involviert sind:

(e-i) In welcher Beziehung stehen die Geschäftsprozesse aus Schritt (b) zu den erhobenen Geschäftsobjekten (Änderung, Erstellung, unveränderliche Verwendung oder Eliminierung)?

Verantwortliche Rolle für die Beantwortung: R-DE

(f) Identifikation der Beziehungen zwischen den Strukturen:

(f-i) Wie werden die erhobenen Geschäftsobjekte aus Schritt (d) zu den durch in Schritt (e) identifizierten Beziehungen zwischen abgebildeten Geschäftsobjekten und deren Geschäftsprozessen mit den in Schritt (b) identifizierten Komponenten in Beziehung gesetzt (Änderung, Erstellung, unveränderliche Verwendung oder Eliminierung)?

(f-ii) Ergeben sich durch diese Beziehungen zwischen den Strukturen und Komponenten neue Beziehungen zwischen anderen Komponenten (z. B.

die Daten-Hoheit³ einer Komponente zieht die Verwendung der Komponente durch eine andere nach sich)?

- (f-iii) Ergeben sich durch diese Beziehungen zwischen den Strukturen und Komponenten neue Beziehungen zwischen anderen Komponenten und den erhobenen Strukturen?

Verantwortliche Rolle für die Beantwortung: R-RE

- (g) Identifikation, ob die Geschäftsobjekte Zustände haben.

(g-i) Haben die in Schritt (c) erhobenen Geschäftsobjekte Zustände, die sich im Lauf der Bearbeitung durch Geschäftsprozesse ändern?

(g-ii) Welche Ausprägung können Zustände haben?

Verantwortliche Rolle für die Beantwortung: R-DE

- (h) Identifikation, wie die Zustände in Strukturen abgebildet werden.

(h-i) Wie werden die in Schritt (g) erhobenen Zustände in Strukturen abgebildet?

(h-ii) Ändert sich durch die Zustände auch die Involviertheit einer Struktur in einer Komponente?

Verantwortliche Rolle für die Beantwortung: R-RE

- (i) Identifikation von Geschäftsprozessen, die diese Zustände ändern bzw. von der Änderung betroffen sind:

(i-i) Welche Geschäftsprozesse ändern die in Schritt (h) erhobenen Zustände?

(i-ii) Welche Geschäftsprozesse werden durch die Änderung dieser Zustände ausgelöst?

Wenn neue Prozesse identifiziert wurden, die softwareseitig unterstützt werden sollen, gehe zu Schritt (b).

Verantwortliche Rolle für die Beantwortung: R-DE

- (j) Identifikation wie Strukturen in Komponenten involviert sind:

(j-i) Wie wirken sich die in Schritt (i) erhobenen (bzw. deren Rolle geänderten) Änderungen der Zustände auf die unterstützenden Komponenten aus Schritt (b) aus?

(j-ii) Ist es notwendig, die Komponenten aus Schritt (i) aufzuteilen, um deren involvierte Strukturen angemessen darzustellen (z. B. damit eine Komponente nicht mit zwei unterschiedlichen Beziehungen mit einem Geschäftsobjekt steht)?

Verantwortliche Rolle für die Beantwortung: R-RE

- (k) Identifikation von Akteuren (Organisationseinheiten, Maschinen oder Personen bzw. Rollen) die die Geschäftsprozesse verantworten und/oder bearbeiten/ausführen:

³ Wir sprechen von Daten-Hoheit einer Komponente bezüglich einer Struktur dann, wenn diese die Erstellung und alle ändernden Zugriffe auf Instanzen der Struktur kapselt.

(k-i) Welche Akteure bearbeiten die in Schritt (a) bzw. (i) identifizierten Geschäftsprozesse?

(k-ii) Welche Akteure verantworten die in Schritt (a) bzw. (i) identifizierten Geschäftsprozesse?

Verantwortliche Rolle für die Beantwortung: R-DE

(l) Identifikation, ob die gefundenen Akteure in Strukturen abgebildet werden müssen.

(l-i) Werden die Akteure im System abgebildet (z. B. wegen geplanter Authentifizierung)?

(l-ii) Stehen die abgebildeten Akteure in einer Beziehung zueinander?

Gehe zu (b), stoppe, wenn alle relevanten Geschäftsprozesse und Komponenten, die diese unterstützten, erwähnt sind.

Verantwortliche Rolle für die Beantwortung: R-RE

(m) Identifikation von Qualitätseigenschaften und Bedingungen, die erfüllt sein müssen:

(m-i) Welche Qualitätseigenschaften müssen Geschäftsprozesse erfüllen, die Auswirkungen auf die unterstützende Software haben könnten?

(m-ii) Welche Qualitätseigenschaften müssen Geschäftsobjekte erfüllen, die Auswirkungen auf die unterstützende Software haben könnten?

(m-iii) Welche Bedingungen (z. B. gesetzliche Vorgaben) müssen Geschäftsprozesse erfüllen, die Auswirkung auf die unterstützende Software haben könnten?

(m-iv) Welche Bedingungen (z. B. gesetzliche Vorgaben) müssen Geschäftsobjekte erfüllen, die Auswirkungen auf die unterstützende Software haben könnten?

Verantwortliche Rollen für die Beantwortung: R-DE und R-RE

Immer wenn Schritt (m) erreicht wurde, ist eine Iteration komplett. Nach jeder Iteration muss die Konsistenz des Basismodells überprüft werden. Der Modellierer sollte in der Lage sein, das erfasste Wissen in seinen eigenen Worten zu wiederholen.

Wie anfänglich in diesem Abschnitt erwähnt und in Abbildung 7.5 dargestellt, werden nach jeder Kompetenzfrage, die zu einer Antwort geführt hat, weitere Schritte ausgeführt. Diese werden in den Abschnitten 7.3.1.1 und 7.3.1.2 im Detail beschrieben.

7.3.1.1 Bedeutungsidentifikation

Durch die Kompetenzfragen werden Begriffe erhoben, die noch keine konsensuelle Bedeutung für die Stakeholder haben müssen. Dies kann auch dann der Fall sein, wenn es sich beim erhobenen Begriff um ein Homonym handelt. Dieser Abschnitt beschäftigt sich mit der Identifikation einer konsensuellen Bedeutung durch Einschrän-

kung des Deutungsrahmens⁴ durch eine Stereotypisierung von Begriffen.

Begriffe können als stereotypisierte Beziehung oder als Entität im Basismodell repräsentiert sein. Die Identifikation der konsensuellen Bedeutung wird mit Hilfe der ontologische Basis festgestellt. Hier wird folgende Vorgehensweise erläutert:

1. **Konsensuelle Bedeutung identifizieren:** Über den Begriff und die Rolle im Basismodell wird diskutiert.
2. **Stereotypisierung:** Die konsensuelle Bedeutung eines Begriffes wird anhand der Konzepte bzw. den Beziehungen der ontologische Basis identifiziert.

Begriffe, für die es kein geeignetes Konzept bzw. keine geeignete stereotypisierte Beziehung gibt, sind (a) entweder nicht einheitlich verstanden worden, oder (b) liegen außerhalb des Kontextes des Basismodells.

Beispiel zur Identifikation einer konsensuellen Bedeutung. Im Rahmen der Beantwortung einer Kompetenzfrage wurde der Begriff *Bestellung* identifiziert. Je nach Diskussion würden sich dafür die Konzepte *BusinessProcess*, *BusinessObject*, usw. eignen. Erkennbar am Beispiel ist, dass die Bedeutung noch nicht klar identifizierbar ist. Dieses Beispiel zeigt, dass der Begriff weiter präzisiert werden muss, um seine Bedeutung identifizieren zu können. Z. B. könnte dies wie im Folgenden geschehen:

- ▶ Bestellerfassung ist ein *BusinessProcess*,
- ▶ Bestellbearbeitung ist ein *BusinessProcess*,
- ▶ Bestellung ist ein *BusinessObject*.

Dieses Beispiel zeigt, dass der Begriff *Bestellung* mehrdeutig ist. In der Literatur [Ste71, MA99] ist es anerkannt, dass die Stereotypisierung (org. Klassifizierung oder Kategorisierung) von Begriffen ein wichtiges Instrument des Menschen zum Verstehen der Bedeutung von Begriffen ist.

7.3.1.2 Identifikation des Zusammenhangs zwischen Entitäten

Dieser Abschnitt beschreibt das Vorgehen zur Identifikation des Zusammenhangs von identifizierten Begriffen bzw. Begriffen, die Beziehungen repräsentieren.

Wir sprechen in diesem Zusammenhang von zurückgelegter Entität bzw. stereotypisierter Beziehung, wenn kein geeigneter Zusammenhang im Rahmen dieses Schrittes identifiziert werden konnte und die Identifikation auf eine der nächsten Iterationen verschoben wird. Ein Zurücklegen bedeutet daher, dass eine Entität bzw. eine stereotypisierte Beziehung in dieser Iteration nicht im Basismodell erfasst wird und die Erfassung auf eine der folgenden Iterationen verschoben wird.

⁴ Der Deutungsrahmen, u.a. identifiziert und beschrieben von Entman [Ent93], bezeichnet die in einem Menschen bereits vorhandene, von externen Einflüssen (gesellschaftlich, individuell) geprägte Wissensstruktur [AS03], die dazu dient, einer aufgenommenen Informationen eine (persönliche) Bedeutung, durch Anreicherung bzw. Interpretation durch bereits vorhandener Wissensstrukturen, zu geben. Die Definition befindet sich auf Seite 28.

Begriff als stereotypisierte Beziehung. Wurde im Rahmen der Erhebung ein Begriff identifiziert, dessen Bedeutung durch eine Beziehung auf Basis der ontologischen Basis repräsentiert wird, werden in diesem Schritt geeignete Entitäten identifiziert, die mittels dieses Begriffs zueinander in Beziehung gesetzt werden.

Folgende Schritte sind vorgesehen:

1. Diskussion über die Bedeutung des Begriffes im Zusammenhang mit anderen Entitäten: Dafür bietet die ontologische Basis Konzepte mit möglichen Beziehungen an, mit deren Hilfe die Entitäten identifiziert werden können, die sich für eine stereotypisierten Beziehungen eignen. Eine Entität eignet sich dann für die Beziehung, wenn das der Entität zugeordnete Konzept den erlaubten Konzepten der Beziehung entspricht.

Beispiel: Die Beziehung *support* ist nur möglich zwischen *Component* und *BusinessProcess*. Die zu identifizierenden Entitäten müssen daher auch jeweils einem dieser Konzepte zugewiesen sein.

2. Identifikation geeigneter Entitäten:
 - ▶ Identifikation von Entitäten, die zurückgelegt wurden und der Bedeutung der Beziehung im Zusammenhang mit Entitäten entsprechen.
 - ▶ Identifikation von Entitäten, die bereits im Basismodell erfasst sind und der Bedeutung der Beziehung im Zusammenhang entsprechen.

Sind keine geeigneten Entitäten vorhanden, werden die stereotypisierten Beziehungen für eine spätere Iteration zurückgelegt.

Begriff ist eine Entität. Wurde im Rahmen der Erhebung ein Begriff identifiziert, dessen Bedeutung durch ein Konzept auf Basis der ontologischen Basis repräsentiert wird, müssen Entitäten identifiziert werden, mit denen der Begriff in Beziehung steht. Folglich müssen auch die stereotypisierten Beziehungen identifiziert werden, die diese Beziehung zwischen der neuen Entität und der bereits im Basismodell vorhandenen Entitäten repräsentiert.

1. Diskussion über die Bedeutung der Entität mit den möglichen Beziehungen zu anderen Entitäten.
2. Identifikation geeigneter Entitäten: In diesem Schritt werden die Entitäten identifiziert, mit denen die zu erfassende Entität in Beziehung stehen soll. Folgende Schritte sind vorgesehen:
 - ▶ Identifikation von Entitäten, die zurückgelegt wurden und mit der die zu erfassende Entität in einer Beziehung stehen soll.
 - ▶ Identifikation von Entitäten, die bereits im Basismodell erfasst sind und mit der die zu erfassende Entität in einer Beziehung stehen soll.

Nach diesem Schritt wird überprüft, ob bereits Beziehungen identifiziert wurden, die den zu erfassenden Zusammenhang repräsentieren. Ist dies nicht der Fall, wird die Entität für die nächste Iteration zurückgelegt, bzw. die Beziehung anhand der Kompetenzfragen identifiziert.

7.3.2 Modellierung: Erweiterung bzw. Anpassung

Dieser Abschnitt behandelt die Erweiterung bzw. Anpassung des Basismodells innerhalb einer einzelnen Iteration.

Dieser Schritt wird von der Rolle des Modellierers durchgeführt. Wir verstehen unter Erweiterung und Anpassung, folgende Aufgaben:

- ▶ Entfernen, Hinzufügen und Ändern von Entitäten und stereotypisierten Beziehungen im Basismodell.

Das Entfernen, Hinzufügen und Ändern von Entitäten und stereotypisierten Beziehungen im Basismodell kann zu umfangreichen Änderungen im Basismodell führen. Z. B. können stereotypisierten Beziehungen zwischen Entitäten aufgrund von Änderungen der zugeordneten Konzepte der Entitäten der Beziehung nicht mehr gültig sein.

Der Modellierer ist verantwortlich dafür, dass das Basismodell nach den ausgeführten Aufgaben weiterhin konsistent ist. Dafür müssen ggf. Entitäten und stereotypisierten Beziehungen aus dem Basismodell entfernt und für spätere Iterationen zurückgelegt werden.

7.3.3 Überprüfung der Konsistenz und Qualität

Bei der Überprüfung der Konsistenz des Basismodells wird sichergestellt, dass die Axiome, spezifiziert in Abschnitt 6.3, gültig sind.

Bei der Überprüfung der Qualität wird die Erfüllung der Qualitätseigenschaften, erläutert in Abschnitt 6.4, kontrolliert.

7.3.4 Fertigstellung des Basismodells

Das Basismodell ist fertig, wenn (a) alle zurückgelegten Entitäten und stereotypisierten Beziehungen im Basismodell erfasst werden konnten, bzw. deren Bedeutung für die erfassten Inhalte der Problem- und Lösungsdomäne geklärt wurden und sich herausgestellt hat, dass diese nicht erfasst werden sollen, (b) das Basismodell konsistent ist und die Qualitätseigenschaften erfüllt sind und (c) wenn die Verknüpfung zu anderen Artefakten möglich ist.

7.4 Verknüpfung zu anderen Artefakten: Integration

Dieser Abschnitt beschreibt beispielhaft die Verknüpfung zu anderen Artefakt-Typen.

Wir berücksichtigen in diesem Abschnitt Artefakt-Typen, die in frühen Phasen des Requirements Engineerings zum Einsatz kommen. Diese entnehmen wir der durchgeführten Studie, siehe dazu Kapitel 3.

Wir zeigen die Verknüpfung eines Basismodells zu den Inhalten konkreter Artefakte dieser Artefakt-Typen.

Zweck der Beschreibung der Verknüpfung ist die Sicherstellung, dass die Artefakte der frühen Phasen des Requirements Engineerings, (a) untereinander durch das Basismodell integrierbar sind und (b) zumindest in Bezug auf das Basismodell untereinander konsistent sind.

7.4.1 Verknüpfung allgemein

Der Modellierungsansatz sieht vor, dass alle im Basismodell enthaltenen Entitäten und stereotypisierten Beziehungen von zumindest einem weiteren Artefakt detailliert werden müssen.

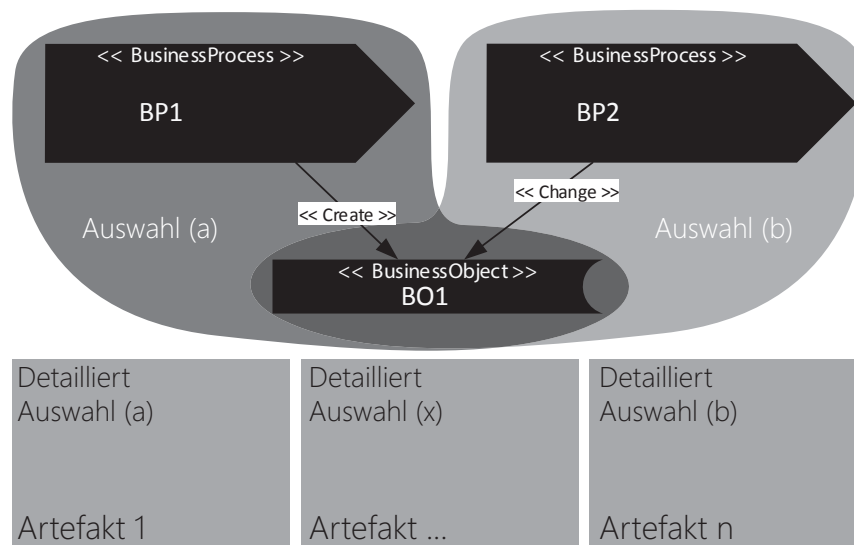


Abbildung 7.8: Allgemeines Detaillieren von Ausschnitten (in der Abbildung als (a) und (b) gekennzeichnet) in weiteren Artefakten

In Abbildung 7.8 sind im oberen Teil Entitäten und Beziehungen abgebildet. Diese entsprechen einem Ausschnitt eines Basismodells.

Die Auswahl (a) besteht aus den Entitäten *BP1* und *BO1* sowie der stereotypisierten Beziehung *Create* zwischen diesen Entitäten. Die Auswahl (b) besteht aus den Entitäten *BP2* und *BO1* sowie der stereotypisierten Beziehung *Change* zwischen diesen Entitäten.

Eine Verknüpfung zu anderen Artefakten besteht immer aus einer Auswahl und einem Artefakt, das diese Auswahl beschreibt. In Abbildung 7.8 sind die beschreibenden Artefakte im unteren Teil der Abbildung eingezeichnet. Ein Ausschnitt kann von beliebig vielen Artefakten beschrieben und typischerweise auch detailliert werden.

Im Ansatz sehen wir vor, dass immer dann, wenn ein Artefakt eine Auswahl des Basismodells detailliert, diese Detaillierung konform zum Basismodell sein muss. Dies bedeutet, dass z. B. eine Detaillierung einer Beziehung im Artefakt nicht widersprüchlich zur Beziehung im Basismodell sein darf.

Beispielsweise wird in Artefakt 1 die Auswahl (a) detailliert. Im Basismodell ist festgelegt, dass *BP1* mit *BO1* in der stereotypisierten Beziehung *Create* steht. Die Detaillierung in Artefakt 1 muss nun diese Beziehung in der Notation des gewählten

Artefakt-Typs abbilden. Das detaillierende Artefakt muss daher explizieren, wie die Erstellung (*Create*) von *BO1* vollzogen wird. Zusätzlich darf die Detaillierung aber keine anderen im Basismodell eingezeichneten Entitäten und Beziehungen detaillieren, die nicht Teil der Auswahl sind.

Eine Detaillierung einer Auswahl des Basismodells in einem Artefakt muss alle in der Auswahl enthaltenen Entitäten und stereotypisierten Beziehungen enthalten und typischerweise detaillieren.

Diese Vorgehensweise stellt sicher, dass alle Artefakte die in frühen Phasen des Requirements Engineerings erstellt werden und auf dem Basismodell basieren, zumindest konsistent in Bezug auf die im Basismodell erfassten Sachverhalte sind. Zusätzlich erreichen wir damit, dass die Abstraktionsebenen der auf dem Basismodell basierenden Artefakte zwar nicht gleich sind, aber durch das Basismodell keine zusätzlichen Interpretationslücken geschlossen werden müssen und dadurch eine Integration der Artefakte z. B. in einer Gesamtsystemspezifikation möglich ist, ohne diese nur informell zu beschreiben.

In den folgenden beispielhaft gezeigten Verknüpfungen zu anderen Artefakten entspricht der dargestellte Ausschnitt auch der Auswahl.

Zur Vollständigkeit der Verknüpfung in den Beispielen. Die in diesem Abschnitt gezeigte Verknüpfung von Ausschnitten eines Basismodells zu anderen Artefakten ist vollständig in Bezug auf die Verwendung der Begriffe und stereotypisierten Beziehungen in den jeweiligen Artefakten. Abhängig vom Artefakt-Typ kann es vorkommen, dass die Verknüpfung nicht vollständig möglich ist. Weiters muss beachtet werden, dass eine Übereinstimmung der Bedeutung einer Detaillierung in einem Artefakt zu einem Ausschnitt eines Basismodells, bei einer Abstraktion der Detaillierung, gegeben ist, sofern das Artefakt genau diesen Ausschnitt des Basismodells detailliert.

7.4.2 Verknüpfung zu Anwendungsfällen

Anwendungsfälle beschreiben die Interaktion der Akteure mit dem (geplanten bzw. realisierten) System. Sie sind eher informell wie formal und häufig als Text (z. B. vorgegeben durch Textschablonen nach Cockburn [Coc07]) gegeben.

Cockburn [Coc07] spricht bei einem Anwendungsfall von einer Übereinkunft zwischen verschiedenen Stakeholdern über das Verhalten eines Systems zu den Akteuren. Jeder Satz in einem Anwendungsfall beschreibt eine Aktion, die ein bestimmtes Stakeholder-Interesse wahrt oder erweitert [Coc07]. Ein Satz kann eine Interaktion zwischen zwei Akteuren beschreiben oder die interne Systemreaktion, die zur Wahrung der Stakeholder-Interessen erforderlich ist [Coc07].

Anwendungsfälle können mit dem Basismodell relativ einfach verknüpft werden. Dazu wird ein Geschäftsprozess gewählt, der die Interaktion mit dem System als Aktivität beinhaltet. Der Geschäftsprozess wird so weit detailliert, bis der Geschäftsprozess nur mehr die Aktivitäten zwischen System und Akteur beinhaltet.

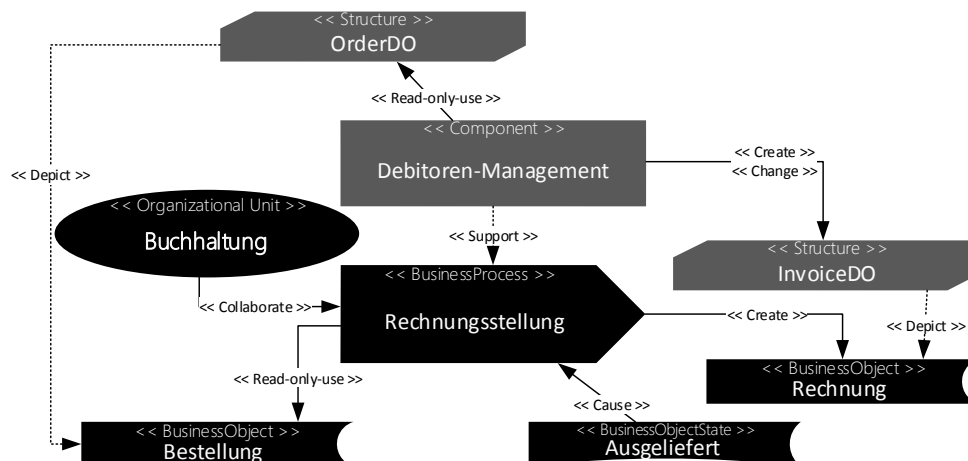


Abbildung 7.9: Ausschnitt aus dem Basismodell aus Abbildung 6.2 zum beispielhaften Verknüpfung zu Anwendungsfällen

Vorbereitung. Vor Erstellung eines Anwendungsfalles wird ein Geschäftsprozess gewählt, der so weit detailliert ist bzw. wird, bis dieser nur mehr die Interaktion mit dem System beschreibt. Die Detaillierung kann im Basismodell wie auch in einem anderen Artefakt geschehen. Je nachdem, was der Anwendungsfall darstellen soll, wird die Struktur⁵ mitbeschrieben. Z. B. wenn das Ziel ist, die Interaktion mit dem System in einer Black-Box-Sicht zu beschreiben, werden die Strukturen nicht berücksichtigt sondern nur die Komponenten, die den Geschäftsprozess direkt unterstützen. Der Ausschnitt wird dann identifiziert indem man, ausgehend vom Geschäftsprozess, die auslösenden Zustände, Geschäftsprozesse und Komponenten in den Ausschnitt aufnimmt und diese bilden die Auslöser im Anwendungsfall. Zusätzlich werden die im Geschäftsprozess involvierten Geschäftsobjekte und verantwortlichen bzw. mitwirkenden Rollen bzw. Personen, Organisationseinheiten bzw. Maschinen ausgewählt. Softwaresystemseitig werden die unterstützenden Komponenten, sowie, wenn keine Black-Box-Sicht beschrieben werden soll, die involvierten Strukturen ausgewählt.

Vorgehen. Wir verwenden zur Beschreibung des Anwendungsfalles die Schablone von Cockburn [Coc07]. Wir gehen in den einzelnen Spalten darauf ein, wie die Informationen aus dem Basismodell gelesen werden können:

Anmerkungen bezüglich Verknüpfung zu den einzelnen Teilen der Schablone in Tabelle 7.1:

Zu (a): Der Name des Anwendungsfalles beinhaltet den Namen des (Teil-)Geschäftsprozesses.

Zu (b): Die Beschreibung leitet sich aus dem Standard- und Alternativverlauf ab.

Zu (c): Die Akteure sind direkt aus dem Ausschnitt des Basismodells in Abbildung 7.9 lesbar. Akteure sind die Organisationseinheiten, Personen oder Rollen, bzw. Maschinen, die die Geschäftsprozesse ausführen oder verantworten. Systemseitig sind

⁵ Der Begriff Struktur wird in Abschnitt 6.2 definiert.

(a)	Name	UCx.x: Rechnungsstellung bei Auslieferung
(b)	Beschreibung	Sobald die Ware ausgeliefert ist, wird die Rechnung von der Buchhaltung erstellt. Dazu wird systemseitig das Debitoren-Management verwendet ...
(c)	Beteiligte Akteure	Buchhaltung (als Akteur), Debitoren-Management (als System)
(d)	Auslöser	Bestellung ist <i>ausgeliefert</i> .
(e)	Vorbedingung	–
(f)	Invarianten	<i>OrderDO</i> bleibt unverändert, <i>Bestellung</i> bleibt unverändert.
(g)	Nachbedingung	<i>Rechnung</i> erstellt und vom System erfasst.
(h)	Standardablauf	<ul style="list-style-type: none"> (i) Der Akteur bekommt die <i>Bestellung</i> zur Abrechnung (Auslösung). (ii) Das System bietet die Möglichkeit, die Bestellung zu suchen. (iii) Der Akteur gibt die entsprechenden Daten ein und bestätigt die Suche. (iv) Das System zeigt die auf der <i>OrderDO</i> basierende Ansicht der Bestellung an. (v) Das System bietet die Möglichkeit, daraus eine Rechnung zu erstellen. (vi) Der Akteur verwendet die Möglichkeit, daraus eine Rechnung zu erstellen und bestätigt dies. (vii) Das System erstellt eine digitale Version der Rechnung <i>InvoiceDO</i> und speichert diese. (viii) Das System zeigt die digitale Version der Rechnung dem Akteur an. (ix) Das System bietet die Möglichkeit, die Rechnung zu drucken. (x) Der Akteur druckt die <i>Rechnung</i>. (xi) Das System bestätigt den Druck und zeigt die Rechnung wieder an. (xii) Der System bietet die Möglichkeit, den Vorgang zu beenden. (xiii) Der Akteur bestätigt das Beenden des Vorganges. (xiv) Das System zeigt den Ausgangsbildschirm.
(i)	Alternativverlauf	<ul style="list-style-type: none"> (iv) Das System zeigt an, dass die Suche nicht erfolgreich war. (v) Das System bietet die Möglichkeit, die Suche zu wiederholen. (vi) Der Akteur bestätigt, dass er die Suche wiederholen möchte. <p>Weiter bei Standardablauf (iii).</p>

Tabelle 7.1: Beispiel-Anwendungsfall abgeleitet aus dem Basismodell

jene Akteure die Komponenten, die die Geschäftsprozesse unterstützen.

Zu (d): Die Auslösung des Anwendungsfalles entspricht der Auslösung des Geschäftsprozesses. Dies kann durch die Anstoßung einer Komponente eines Geschäftsprozesses bzw. der Änderung eines Zustandes erfolgen.

Zu (e): Vorbedingungen können z. B. aufgrund der involvierten Geschäftsobjekte oder deren Abbildung abgeleitet werden, z. B. je nach Abstraktionsebene kann angegeben werden, dass die *OrderDO* eine gültige Bestellung repräsentieren muss.

Zu (f): Invarianten lassen sich durch die Art der Involviertheit von Geschäftsobjekten in Geschäftsprozesse sowie durch die Involviertheit von Strukturen in Komponenten bestimmen. Z. B. im Rahmen eines Anwendungsfalles darf eine Struktur nicht verändert werden, da diese nur rein lesend (*Read-only-use*) im Basismodell verwendet wird.

Zu (g): Die Nachbedingungen kann, wie die Vorbedingung, z. B. durch die Art der Involviertheit (Änderung, Erstellung, Eliminierung) von Geschäftsobjekten in Geschäftsprozessen oder von Strukturen in Komponenten bestimmt werden.

Zu (h): Der Ablauf darf nicht im Widerspruch zum im Basismodell dargestellten Sachverhalt stehen. Z. B. wenn ein Geschäftsobjekt nur lesend in einem Geschäftsprozess involviert ist, darf dieser im Anwendungsfall nicht schreibend involviert sein.

Zu (i): Der Alternativverlauf detailliert die Beziehung zwischen dem System und dem Akteur und steht in keinem Widerspruch zum Basismodell.

7.4.3 Verknüpfung zu Sequenzdiagrammen

Unser in Abbildung 7.10 ausgewählter Ausschnitt des Basismodells aus Abbildung 6.2 zeigt bekanntlich eine Struktursicht auf die Problem- und Lösungsdomäne. In Sequenzdiagrammen werden typischerweise die Interaktionen zwischen Objekten dargestellt. Dabei ist das Hauptaugenmerk darauf gelegt, die Reihenfolge der Interaktionen mit involvierten *Daten* darzustellen.

Laut Rumpe [Rum11] ist ein Sequenzdiagramm grundsätzlich exemplarisch. Die dargestellten Informationen können daher während des Ablaufes eines Systems beliebig häufig auftreten [Rum11]. Rumpe [Rum11] weist weiters darauf hin, dass aufgrund der Exemplarizität Sequenzdiagramme nicht zur vollständigen Modellierung von Verhalten geeignet sind und vor allem während der Anforderungsdefinition [...] eingesetzt werden.

Wir verwenden die Sequenzdiagramme zur Detaillierung von Beziehungen und der Bestimmung der Reihenfolge von Aktivitäten innerhalb der dargestellten Komponenten bzw. Geschäftsprozessen. Wir weisen darauf hin, dass im Rahmen der Darstellung keine Rücksicht auf eine Instanziierungs-Ebene genommen wird. Z. B. steht der Kunde als Person bzw. Rolle in Abbildung 7.10 für die Menge der Kunden, die mit unserem dargestellten System interagieren können. Im Sequenzdiagramm wird diese Menge von Kunden ebenso Kunde genannt, dabei ist im Sequenzdiagramm eine konkrete Instanz eines Kunden gemeint, der mit dem System interagiert.

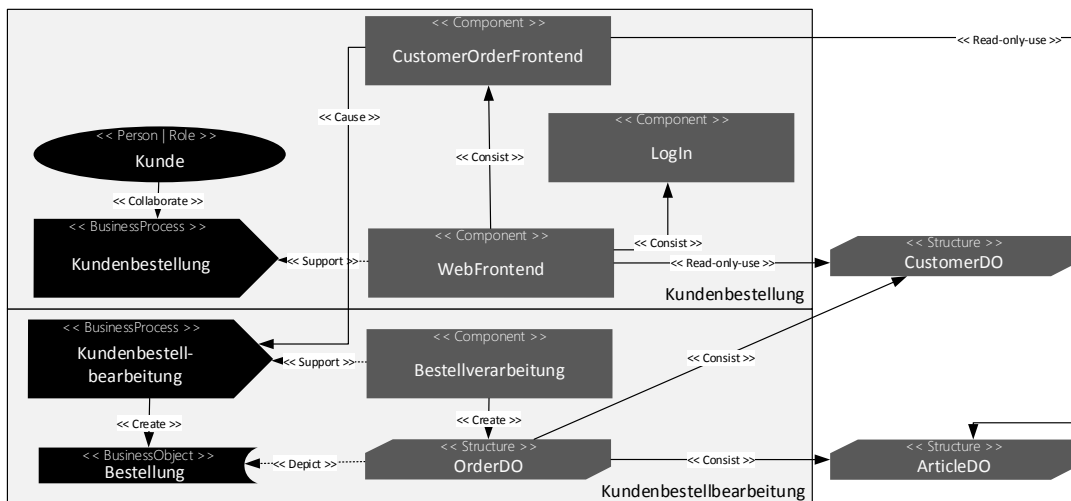


Abbildung 7.10: Ausschnitt aus dem Basismodell aus Abbildung 6.2 zum beispielhaften Verknüpfung zu Sequenzdiagrammen

Vorgehen. Ausgehend von dem darzustellenden Ausschnitt werden z. B. Komponenten gewählt mit allen involvierten Strukturen. Die Interaktion des Systems mit der Umgebung wird mit Hilfe von Geschäftsprozessen ausgedrückt. Ausgehend von der Kundenbestellung ist die Komponente *WebFrontend* involviert, da sie diesen Geschäftsprozess unterstützt. Diese Komponente besteht aus weiteren Komponenten, *CustomerOrderFrontend* und *Login*. In unserer Darstellung wird mit dem *WebFrontend* nur über seine Teilkomponenten interagiert. Im Rahmen der Ausführung wird ein Geschäftsprozess angestoßen, nämlich *Kundenbestellbearbeitung*. Diese drei Komponenten und der Geschäftsprozess dienen uns als Objekte im Sequenzdiagramm. Wie in Abbildung 7.11 und 7.12 zu sehen, werden alle Strukturen und Geschäftsobjekte als Parameter der Interaktionen abgebildet.

Die Verknüpfung von der Kundenbestellung (in Abbildung 7.11) bis hin zu der Kundenbestellbearbeitung (in Abbildung 7.12) wird durch die Nennung des Geschäftsprozesses expliziert, in dessen Rahmen die Bearbeitung durchgeführt wird.

Detaillierung der Cause-Beziehung. In Abbildung 7.11 wird der Geschäftsprozess *Kundenbestellbearbeitung* aus Abbildung 7.10 eingeführt. Die Kundenbestellverarbeitung wird in Abbildung 7.12 detailliert. Die *Cause*-Beziehung zwischen den Komponenten *CustomerOrderFrontend* und dem Geschäftsprozess *Kundenbestellbearbeitung* wird durch die Art der Anstoßung (*Bestellen*) präzisiert. Die von der anstoßenden Komponente verwendeten Strukturen, werden bei der Anstoßung des Geschäftsprozesses als Parameter verwendet.

Abbildung 7.12 zeigt, wie die geforderte Beziehung *Create* zwischen *Kundenbestellbearbeitung* und *Bestellung* aus Abbildung 7.10 im Sequenzdiagramm präzisiert wird. Im Rahmen der Ausführung des Geschäftsprozesses *Kundenbestellbearbeitung* leistet die Komponente *Bestellverarbeitung* Unterstützung. Die real existierende *Bestellung* wird von der Komponente, im Rahmen des unterstützten Geschäftsprozesses, gedruckt. Dies führt uns weiters zur Aussage, dass im dargestellten Beispiel die *Bestellung* nach der Erzeugung von *OrderDO* erstellt wird.

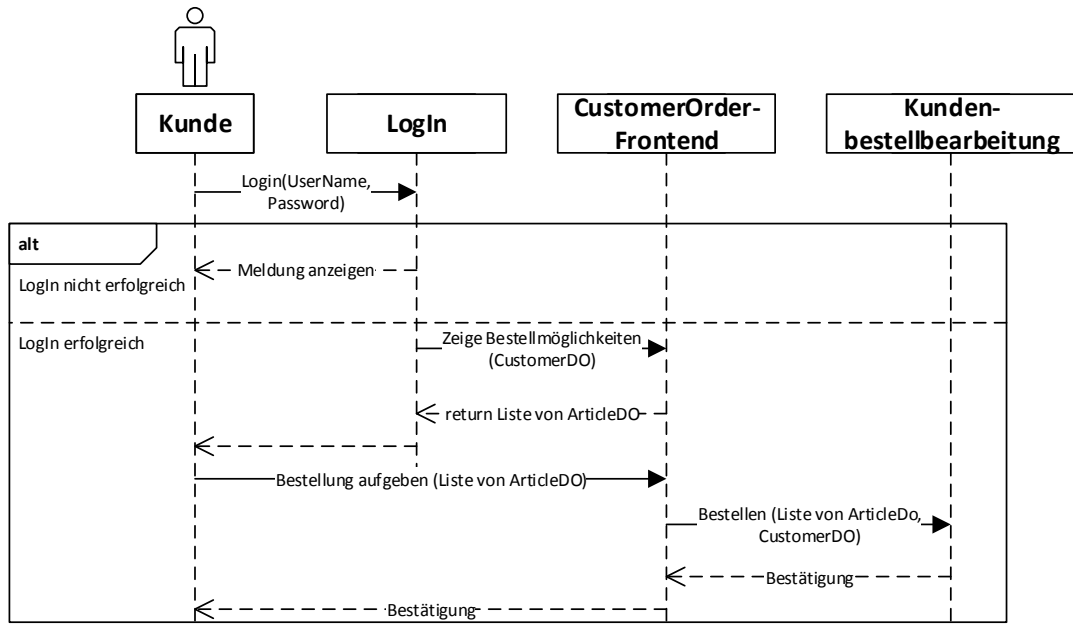


Abbildung 7.11: Sequenzdiagramm, erstellt aus dem Ausschnitt Kundenbestellung, aus Abbildung 7.10

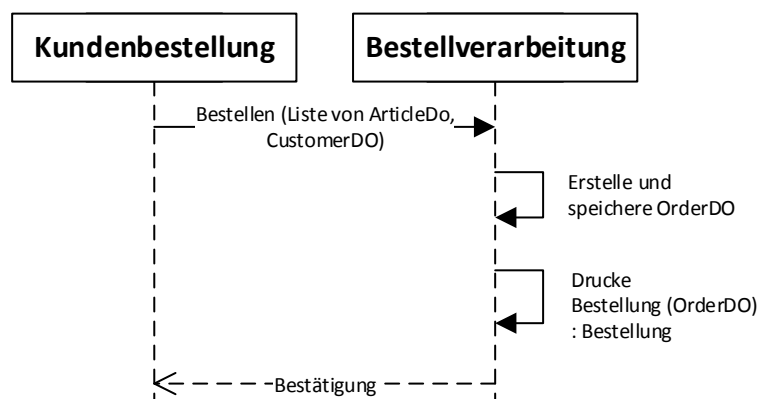


Abbildung 7.12: Sequenzdiagramm, erstellt aus dem Ausschnitt Kundenbestellbearbeitung, aus Abbildung 7.10

7.4.4 Verknüpfung zu Datenmodellen

Wir verwenden in diesem Abschnitt die UML-Klassendiagramm-Notation. Klassendiagramme gehören zu den Strukturdiagrammen der UML. Klassendiagramme werden in einer Vielzahl von Aufgaben im Softwareentwicklungsprozess eingesetzt. Z. B. werden Klassen in der Modellierung der Struktur des Programmcodes verwendet – Klassen werden daher zur Darstellung von strukturellen Zusammenhängen eines System eingesetzt. Darüber hinaus werden sie in der Analyse, im Design und Implementierung eingesetzt [Rum11].

Wir beschäftigen uns in diesem Abschnitt mit der Frage, wie die Verknüpfung vom Basismodell zu einem Datenmodell, modelliert in der UML-Klassendiagramm-Notation, vorgenommen werden kann. Datenmodelle lassen sich erstellen, sobald die Anforderungen an das System bekannt sind. Diese können u.a. aus den in Abschnitt 7.4.2 erstellten Anwendungsfällen gelesen werden.

Grundlegendes. Die Entitäten in Datenmodellen bilden ausgewählte Inhalte von in der Realität existierenden Gegenständen ab. Darüber hinaus beinhalten sie häufig auch Informationen, die nicht im Original vorhanden sind (z. B. eindeutige Identifikatoren in technischen Datenmodellen). In dem in dieser Arbeit vorgestellten Modellierungsansatz modellieren wir diese zwei unterschiedlichen Domänen im Basismodell explizit: (a) Geschäftsobjekte bilden einen Ausschnitt der Realität ab, der explizit nur die Informationen enthält, die auch in der Realität vorhanden sind. (b) Strukturen bilden Geschäftsobjekte ab, die auch bereits lösungsspezifische Informationen enthalten können. Dieses Vorgehen ermöglicht es uns, eine Unterscheidung zwischen dem typischen Verstehen des Problems (der Analyse) mit Modellierung u.a. der Struktur des Problems in Form der Geschäftsobjekte zu betreiben und andererseits die Modellierung der lösungsspezifischen Abbildung des Ausschnittes für die Repräsentation im System durchzuführen.

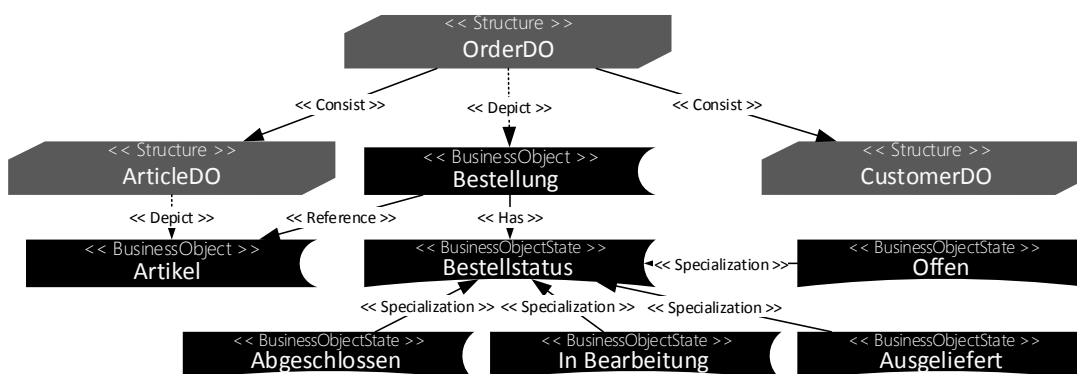


Abbildung 7.13: Ausschnitt aus dem Basismodell aus Abbildung 6.2

Vorgehen. Abbildung 7.13 zeigt einen Ausschnitt aus dem Basismodell aus Abbildung 6.2. Dieser Ausschnitt enthält (a) lösungsspezifische Strukturen mit deren noch generischen Beziehungen untereinander und (b) die problemspezifischen Geschäftsobjekte mit Zustände und deren Beziehungen.

Ausgehend von den Strukturen im Basismodell bilden wir Klassen. Dies zeigt Abbildung 7.14. Damit werden anfänglich drei Klassen erstellt, *OrderDO*, *ArticleDO* und *CustomerDO*. Die Attribute der Klasse müssen aus den Anforderungen an das System noch weiter bestimmt werden. Dies gilt auch für die Kardinalitäten der Referenzen. Dies kann geschehen, indem z. B. die Anwendungsfälle, Sequenzdiagramme usw., die im Rahmen der Verknüpfung erstellt wurden, betrachtet werden. In unserem Fall geben uns die im Rahmen der *Kundenbestellung* und *Kundenbestellbearbeitung* erstellten Sequenzdiagramme, siehe dazu Abschnitt 7.4.3, Hinweise zu den Kardinalitäten.

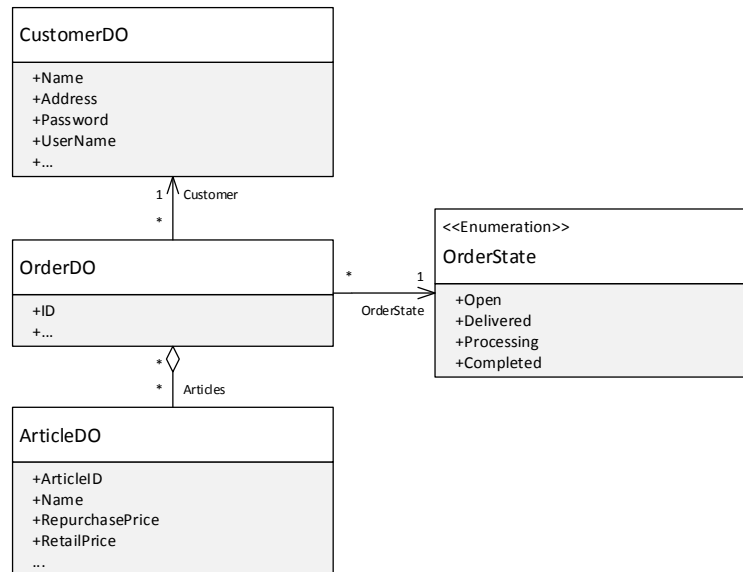


Abbildung 7.14: Klassendiagramm, basierend auf den Ausschnitt aus Abbildung 7.13

Die Struktur *OrderDO* bildet das Geschäftsobjekt *Bestellung* ab, zu dem eine Reihe von Zuständen gehört. Auch diese müssen, bei Verwendung durch die Unterstützung eines Geschäftsprozesses durch eine Komponente, abgebildet werden. Dies ist in Abbildung 6.2 durch den Geschäftsprozess *Rechnungsstellung* gegeben, der durch die Komponente *Debitoren-Management* unterstützt wird. Zusätzlich ist diese Situation durch die Komponente *Auslieferungsmanagement* gegeben, die den Geschäftsprozess *Auslieferung* unterstützt. Dieser Geschäftsprozess verändert den Zustand des Geschäftsobjektes *Bestellung*.

Wie die Zustände in einem Datenmodell abgebildet werden, unterliegt dem Modellierer des Datenmodells. In unserem Fall in Abbildung 7.14 haben wir uns dafür entschieden, die Zustände als Enumeration abzubilden.

7.4.5 Verknüpfung zu Komponentendiagrammen

Auf Basis des Basismodells kann eine Grobarchitektur erstellt werden. Dabei können die Komponenten als Basis für die Strukturierung im Komponentendiagramm 1:1 übernommen werden. Die Beziehungen können auf Basis der involvierten Strukturen identifiziert und daraus auch die Schnittstellen spezifiziert werden.

Wir möchten hier darauf hinweisen, dass diese Vorgehensweise, wie bei allen in diesem Kapitel vorgestellten Vorgehensweisen, nicht angewendet werden muss und nur

beispielhaft ist. Häufig gibt es Gründe, z. B. die Verwendung von Architektur-Mustern, die Architektur anders zu strukturieren.

Wir zeigen in diesem Abschnitt, wie eine Dreischichten-Architektur aus den Komponenten des Basismodells erstellt werden kann. Dabei werden die einzelnen Komponenten einer oder mehreren Schichten durch die Verwendung von Strukturen zugeordnet. Diese Vorgehensweise ist exemplarisch.

Wie auch in den anderen Verknüpfungen in diesem Kapitel erwähnt, müssen die Anforderungen, bei der Identifikation von Komponenten, berücksichtigt werden. Wir bilden eine strikte Schichtenarchitektur. Das bedeutet, einzelne Schichten dürfen nur Abhängigkeiten zur direkt darunterliegenden Schicht aufweisen.

Vorgehen. Wir halten uns an eine klassische Dreischichten-Architektur [Rät98]. Abbildung 7.15 zeigt initial die folgenden Komponenten: Präsentationsschicht, Geschäftslogik und Datenhaltung.

Wir gehen von den Komponenten aus, die Geschäftsprozesse unterstützen und diese Unterstützung eine Interaktion mit Personen bzw. Rollen oder Organisationseinheiten involviert. In unserem Falle wurde das im Beispiel Anwendungsfall und Sequenzdiagramm präzisiert. Für mindestens diese Komponenten muss daher ein Frontend vorgesehen werden.

Durch die strikte Dreischichten-Architektur ist es notwendig, Geschäftslogik-Komponenten einzuführen, da die Geschäftslogik nicht Teil des Frontends sein darf und ein direkter Zugriff auf die Datenhaltung über die Geschäftslogik-Schicht nicht erlaubt ist.

Die Datenhaltung führt Datenzugriffskomponenten ein. Diese haben die Aufgabe, die in der Geschäftslogik eingesetzten Datenobjekte zu verwalten. Als Datenobjekte können die Klassen und deren Beziehungen verwendet werden, die in Abschnitt 7.4.4 erstellt wurden.

Die Abhängigkeiten zwischen den Komponenten der Geschäftslogik und der Datenhaltung identifizieren wir durch die Analyse der Beziehungen zwischen den Komponenten und Strukturen im Basismodell (siehe dazu Abbildung 6.2).

Durch den erstellten Anwendungsfall, den Sequenzdiagrammen und des Datenmodells, lassen sich im Komponentendiagramm schon einige der Abhängigkeiten präzisieren. Wir haben aus Gründen der Übersichtlichkeit diese Schnittstellen im Komponentendiagramm in Abbildung 7.15 nicht präzisiert. Z.B. wissen wir aus dem Sequenzdiagramm, welche Interaktionen auf jeden Fall in der Schnittstelle zwischen *CustomerOrderFrontend* und *LogInFrontend* abgebildet sein müssen.

Beispiel der Detaillierung einer Schnittstelle. Im Anwendungsfall in Abschnitt 7.4.2 haben wir die Suche nach Bestellungen durch die Buchhaltung erfasst. Daraus abgeleitet haben wir das Vorhandensein eines Frontends, das diese Suche ermöglicht. Wir haben in unserem Anwendungsfall die Suche nicht weiter präzisiert, daher nehmen wir an, es handelt sich um die Suche mit Hilfe einer Identifikationsnummer. Die Schnittstelle der Komponente *Debitoren-Management* weist daher zumindest eine Schnittstelle *Suche* auf, die zumindest die Möglichkeit der Suche über die Identifika-

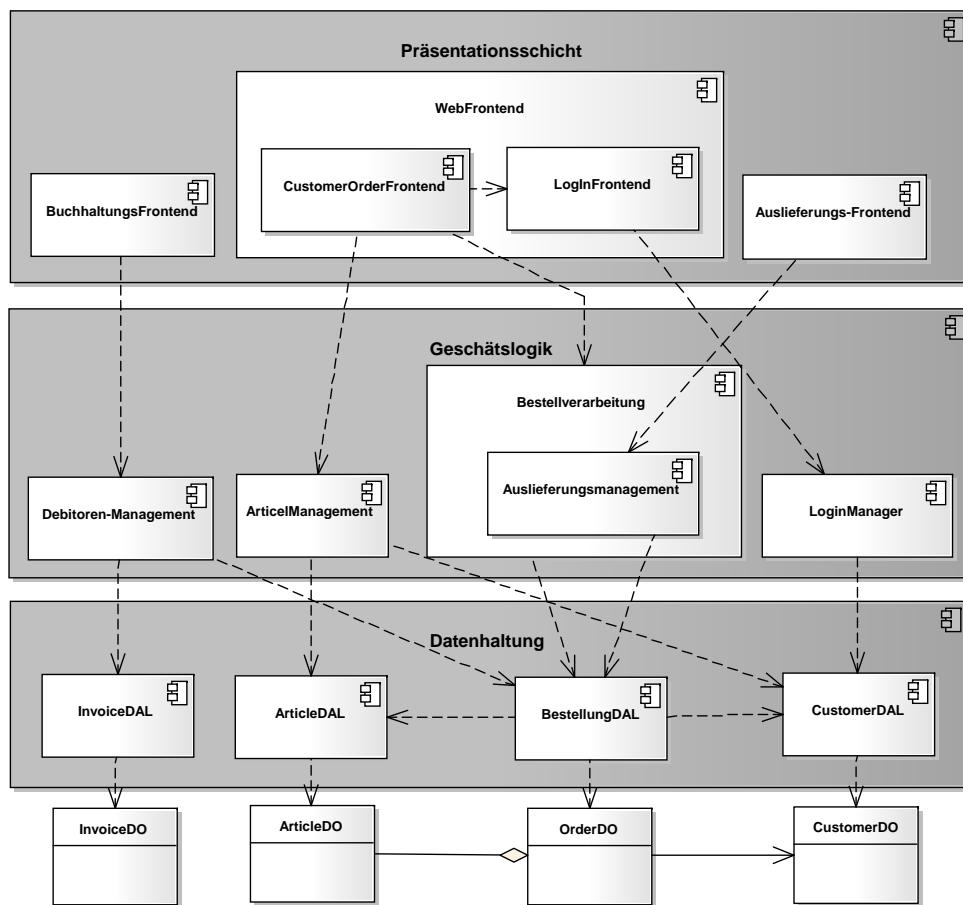


Abbildung 7.15: Beispiel einer Dreischichtenarchitektur mit den abgeleiteten Komponenten aus dem Basismodell aus Abbildung 6.2 und den Anforderungen in diesem Abschnitt

tionsnummer enthält und ein Datenobjekt vom Typ *OrderDO* zurückliefern muss.

Verantwortlichkeiten und Abhängigkeiten zwischen den Datenzugriffskomponenten. Je nach Design-Entscheidung, haben die Datenzugriffskomponenten (dargestellt mit dem Postfix *DAL*) die Aufgabe, den Datenzugriff auf z. B. eine Datenbank zu verwalten. Die Abhängigkeiten zwischen den einzelnen Datenzugriffskomponenten ergeben sich auf Basis der Strukturen im Basismodell. So müssen für das Erstellen der *OrderDO* auch *ArticleDO*s und ein *CustomerDO* erstellt bzw. abgefragt werden. Für die Rechnung (*InvoiceDO*) ist das nicht notwendig, da sich bei der Rechnungserstellung, spezifiziert durch das Sequenzdiagramm in Abschnitt 7.4.3 herausgestellt hat, dass es keinen direkt Bezug der Rechnung zur Bestellung gibt.

7.4.6 Verknüpfung zu Organisationsdiagrammen

Ein Organisationsmodell ist nach [Sch12], die sich bei der Kompetenzabgrenzung auf Hierarchieebene einer Unternehmung ergebende Organisationsstruktur. Nach [Bru10] enthält ein Organisationsmodell die Definition von organisatorischen Rollen,

Funktionen und Stellen im Unternehmen, mit ihrem Bezug zu den Funktionen (Aufbauorganisation), Ablaufschritten (Prozessorganisation) und mit ihrer Beziehung zu Partnern außerhalb des Unternehmens.

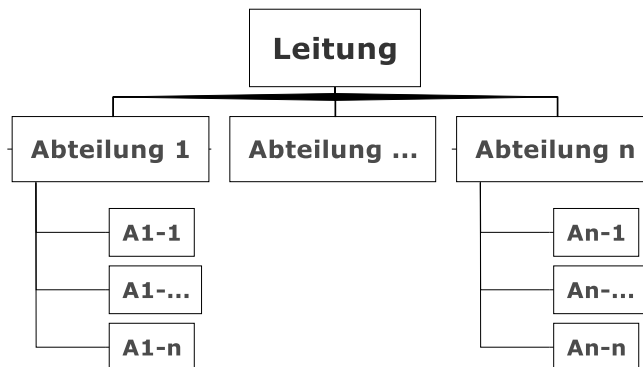


Abbildung 7.16: Einliniensystem-Organigramm

Organisationsstrukturen sind typischerweise nach Weisungsbefugnissen strukturiert. So ist im Einliniensystem jede untergeordnete Instanz jeweils nur einer übergeordneten Stelle unterstellt (siehe Abbildung 7.16).

Abbildung 7.17 zeigt einen Ausschnitt aus dem Basismodell in Abbildung 6.2. Der Ausschnitt zeigt Organisationsstrukturen und Rollen bzw. Personen. Die Organisationseinheiten (*Organizational Unit*) entsprechen dabei den typischen Abteilungen in Organisationsdiagrammen. Die Rollen bzw. Personen entsprechen den Stellen.

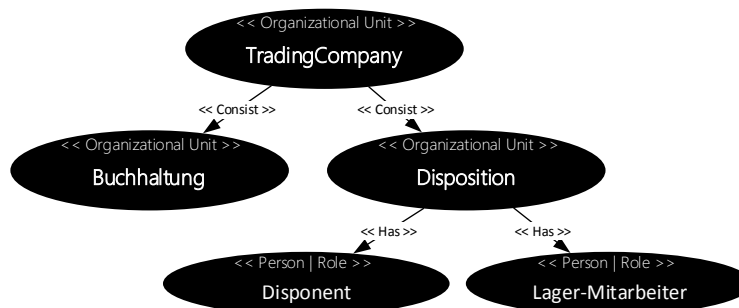


Abbildung 7.17: Organisationseinheiten und Rollen bzw. Personen im Basismodell

Im Basismodell können Organisationseinheiten aus anderen Organisationseinheiten bestehen (*Consist*). Stellen bzw. Rollen werden den Organisationseinheiten zugeteilt (*Has*). In unserem Ansatz ist es möglich, ein Mehrliniensystem zu realisieren. In einem Mehrliniensystem kann es mehrere übergeordnete Stellen geben. Wobei wir als Stelle eine Organisationseinheit betrachten und Personen bzw. Rollen mehreren Organisationseinheiten zugeordnet sein können. Organisationseinheiten können aber immer nur Teil einer übergeordneten Organisationseinheit sein.

Vorgehen. Im Basismodell werden die Organisationseinheiten und Rollen bzw. Personen erfasst, die in Geschäftsprozesse involviert sind. Im Konstruktionsprozess wird angeführt, dass die in Geschäftsprozessen involvierten Akteure identifiziert werden. Im Anschluss an die Identifikation werden diese in Beziehung gesetzt.

Umgekehrtes Vorgehen. Organigramme sind häufig bereits im Unternehmen vorhanden, wenn Requirements Engineering für ein Softwareprojekt betrieben wird. Diese Organigramme sind typischerweise umfangreicher als die Ausschnitte, die benötigt werden, um die Organisationseinheiten und Rollen bzw. Personen zu identifizieren, die in den Geschäftsprozessen, die für ein Softwaresystem relevant sind, zu beschreiben. Im Basismodell erfasst werden dann die Organisationseinheiten und Rollen bzw. Personen, die in den Geschäftsprozessen involviert sind. Abschließend werden die Organisationseinheiten und Rollen bzw. Personen in Beziehung gesetzt. Es findet vom vorhandenen Organigramm daher eine Auswahl statt.

7.4.7 Verknüpfung zu Geschäftsprozessdiagrammen

Um Geschäftsprozesse zu verwalten, werden diese beschrieben und dokumentiert [All09]. Im einfachsten Fall werden dazu textuelle oder tabellarische Beschreibungen verwendet. Häufig werden auch Präsentations- oder Grafikprogramme verwendet, um einfache Ablaufdiagramme zu erstellen. Dann bestehen sie aus Kästchen und Pfeilen, wobei keiner bestimmten Methodik gefolgt wird [All09].

Beispielhaft wird hier die Verknüpfung zu ereignisgesteuerter Prozessketten⁶ gezeigt. Die ereignisgesteuerten Prozessketten bilden den Kern der ARIS-Methode. ARIS steht für Architektur integrierter Informationssysteme. Die EPKs bilden die zentrale Beschreibungssprache der ARIS-Methode [Sch02, Leh07]. In der Anwendung ist ARIS streng mit dem Werkzeug der IDS Scheer AG verbunden, dadurch gibt es keine Organisation, die die Beschreibungssprache verbindlich vorgibt und/oder weiterentwickelt – es fehlt daher ein Standard [All09, Mor10].

Vorgehen. Beginnend bei Geschäftsprozessen, die aus anderen Geschäftsprozessen bestehen, werden die Einzelaktivitäten innerhalb der übergeordneten Geschäftsprozessen erfasst. Abbildung 7.18 zeigt einen Ausschnitt des Basismodells aus Abbildung 6.2.

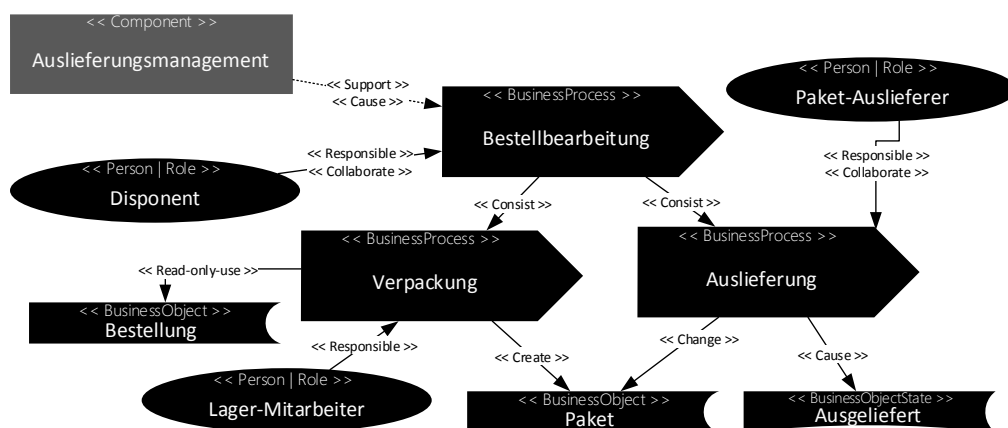


Abbildung 7.18: Ausschnitt aus dem Basismodell

⁶ Erweiterte ereignisgesteuerte Prozessketten (EPPK) ist eine Sprache zur Modellierung von Geschäftsprozessen und Arbeitsabläufen und ist wichtiges Element des ARIS-Konzeptes (Architektur Integrierter Informationssysteme) [SN00].

Der Geschäftsprozess *Bestellbearbeitung* besteht aus zwei Geschäftsprozessen, *Verpackung* und *Auslieferung*. Die Geschäftsprozesse, aus denen der übergeordnete Geschäftsprozess besteht, müssen in den Einzelschritten expliziert werden. In Abbildung 7.19 wird der Geschäftsprozess *Bestellbearbeitung* in Einzelschritten beschrieben. Hierbei ist zu beachten, dass die im Basismodell explizierten Rollen und Geschäftsobjekte im Geschäftsprozessmodell vorkommen. Die Rolle *Disponent* kommt im Geschäftsprozess bei den Einzelschritten in Abbildung 7.19 als Rolle vor.

Die Reihenfolge der Ausführung der Geschäftsprozesse, aus denen der Geschäftsprozess *Bestellbearbeitung* besteht, wird im Geschäftsprozessmodell detailliert.

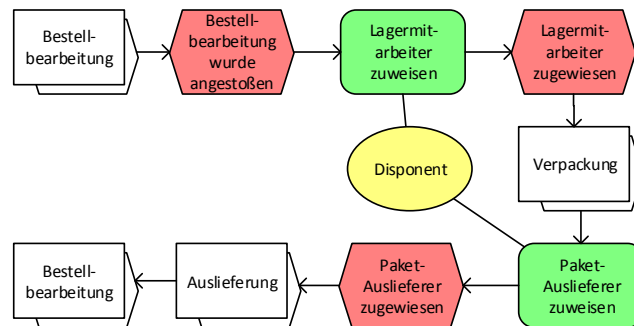


Abbildung 7.19: Geschäftsprozess *Bestellbearbeitung* in Einzelschritten

Abbildung 7.20 zeigt die Geschäftsprozesse *Verpackung* und *Auslieferung* in Einzelschritten. Die im Basismodell in Abbildung 7.18 angeführten Geschäftsobjekte und Rollen kommen auch in diesen Modellen vor. Die Erstellung des Paketes wird im Geschäftsprozess *Verpackung* nur durch die Bezeichnung der Aktivität angedeutet, im Basismodell geschieht dies über die Rolle der Involviertheit des Geschäftsobjektes in den Geschäftsprozess explizit.

Alle Geschäftsprozessmodelle dürfen dem Basismodell nicht widersprechen. Z. B. würde das Paket in Geschäftsprozess *Verpackung* anders involviert sein, als es im Basismodell ist, würde ein Widerspruch vorliegen. Dann muss entweder das Basismodell oder das Geschäftsprozessmodell geändert werden. Wird das Basismodell geändert, müssen alle davon abhängigen Artefakte identifiziert werden und ggf. abgeändert werden.

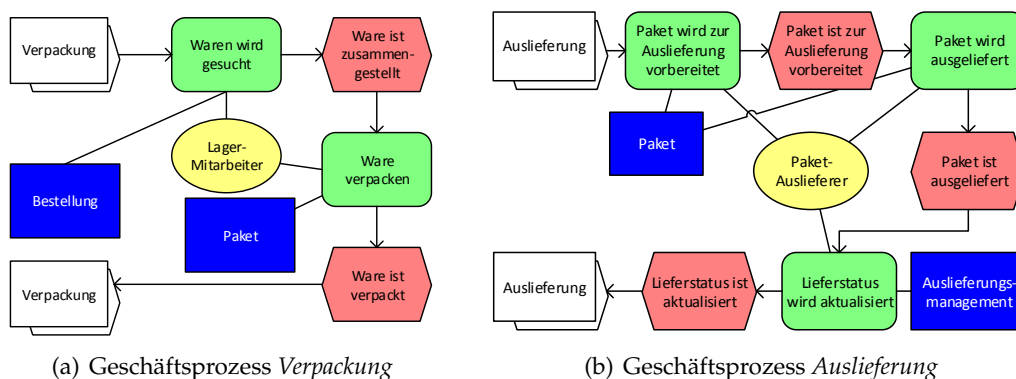


Abbildung 7.20: Die Geschäftsprozesse *Verpackung* und *Auslieferung* in Einzelschritten

Umgekehrtes Vorgehen. Artefakte, die Geschäftsprozesse beschreiben, sind häufig schon vor der Erstellung eines Basismodells vorhanden. Geschäftsprozesse sollten nur ein Geschäftsobjekt involvieren. Wenn dies nicht möglich ist, soll der Geschäftsprozess weiter aufgeteilt werden. Dies ist wichtig, um die Art der Unterstützung von Komponenten für die Verknüpfungen zu anderen Artefakt-Typen besser herauslesen zu können. Hier sei angemerkt, dass dies nur eine Richtlinie ist.

7.4.8 Verknüpfung zu allgemeinen SOA-Ansätzen

Nach [OAS06] ist eine serviceorientierte Architektur (SOA) ein Paradigma für die Organisation bzw. Strukturierung und Verwendung von verteilter Funktionen, die unter der Kontrolle von verschiedenen Anbietern stehen können. SOA ist ein Paradigma, daher ein (Denk-)Muster.

Mit SOA rücken Workflows in das Zentrum der Betrachtung. Workflows sind weitgehend automatisierte Abläufe, deren Akteure Maschinen sind und nicht mehr Menschen [Mel08]. Wird ein Workflow von einem nichttechnischen Standpunkt aus betrachtet, spricht man von einem Geschäftsprozess [Mel08].

Im Modellierungsansatz dieser Arbeit können Maschinen in einen Geschäftsprozess und daher auch in einen Workflow involviert sein. Dies reicht von der Spezifikation der Verantwortung, die eine Überwachung bedeutet, bis zur Ausführung. Das heißt, dass ein Geschäftsprozess bzw. Workflow von einer Maschine ausgeführt wird, kann im Basismodell ausgedrückt werden.

Abbildung 7.21 zeigt ein wichtiges Konzept von SOA: Die lose Kopplung. Softwareelemente (darunter versteht man alle Einheiten, die Funktionen enthalten, z. B. Komponenten) in SOA müssen Softwareelemente finden, die ihren Anforderungen genügen. Hierzu wird typischerweise in Dienstverzeichnissen gesucht. Dienstverzeichnisse sind Verzeichnisse, die die Beschreibung von Fähigkeiten von zur Verfügung gestellten Diensten (also nutzbaren Softwareelementen) enthalten. Wird ein Dienst im Verzeichnis gefunden, der die Anforderungen erfüllt, wird dieser für diese Aufgabe verwendet.

Diese lose Kopplung soll dafür sorgen, dass Dienstanbieter theoretisch schnell ersetzt werden können. Ein Softwaresystem, das auf SOA basiert besteht aus einer Struktur voneinander isolierbaren Elemente, die durch andere Elemente ausgetauscht werden können [Mel08].

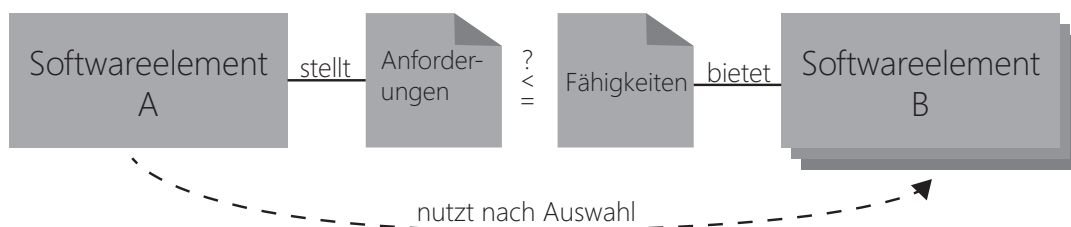


Abbildung 7.21: Lose Kopplung von Softwareelementen, angelehnt an [Góm12]

In SOA wird häufig auch eine feste Kopplung verwendet. Die verwendeten Dienste sind dann nicht über ein Dienstverzeichnis auffindig gemacht worden, sondern

wurden direkt angebunden.

Kann das Basismodell bei Verknüpfungen zu SOA-Architekturen helfen? Die Komponenten des Basismodells können als Anforderungen an das System gesehen werden. Qualitätseigenschaften und andere Bedingungen können im Basismodell erfasst werden. Die auf dem Basismodell aufbauenden Artefakte, die im Rahmen dieses Abschnittes behandelt wurden, sind Anforderungen an das System. Die Anforderungen an einzelne Komponenten (bzw. Dienste) können daher auf Basis des Basismodells ermittelt werden.

In SOA spielt die Definition der Schnittstelle eine besondere Rolle. Es muss sichergestellt werden, dass die über die Schnittstellen kommunizierten Daten auch entsprechend der Anforderungen strukturiert sind und die Bedeutung der Inhalte mit den Anforderungen übereinstimmt. Wie in Abschnitt 7.4.4 kurz erläutert, sehen wir die Datenmodelle, die die zwischen den Komponenten zu kommunizierenden Typen spezifizieren, als wichtige Basis für die Schnittstellenspezifikation. Das Basismodell bietet die Möglichkeit, früh im Requirements Engineering diese Typen zu benennen und später, wie in Abschnitt 7.4.4 beschrieben, zu detaillieren.

In Abschnitt 7.4.5 wurde erläutert, wie auf Basis der Anforderungen und des Basismodells Schnittstellen abgeleitet werden können. Diese können im Rahmen von SOA verwendet werden.

7.5 Zusammenfassung

In diesem Abschnitt wurde beschrieben, wie der Modellierungsansatz methodisch eingesetzt werden kann. Die Methodik wurde in Abschnitt 7.1 in den Softwareentwicklungsprozess und in die Artefaktorientierung eingeordnet.

Die an der Erstellung des Basismodells beteiligten Rollen wurden in Abschnitt 7.2 eingeführt. Darauf aufbauend wurde der Konstruktionsprozess in Abschnitt 7.3 beschrieben. Abschließend wurde in Abschnitt 7.4 beispielhaft die Verknüpfung zu anderen Artefakt-Typen aufgezeigt.

Evaluation

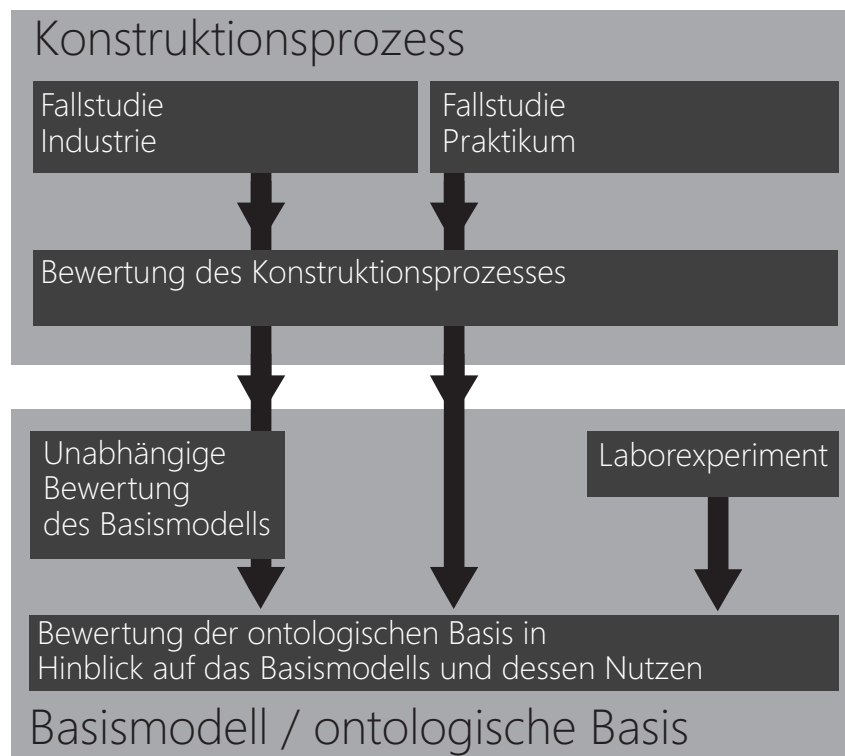


Abbildung 8.1: Vorgehen zur Evaluierung des Ansatzes

Zur Evaluierung des Ansatzes wurden Aktionsforschungs-Studien sowie ein Labor-experiment durchgeführt.

In Fallstudie 1 wurde der Ansatz im Umfeld eines Industrieprojektes angewendet. In Fallstudie 2 wurde der Ansatz im Umfeld von App-Entwicklungsprojekten angewendet. Beide Studien werden in Abschnitt 8.1 vorgestellt.

Ziel der Fallstudien 1 und 2 ist die Bewertung der Anwendbarkeit des Konstruktionsprozesses, sowie die Eignung des Basismodells als Basis für weitere Artefakte in frühen Phasen des Requirements Engineerings.

Das erstellte Modell aus Fallstudie 1 wurde von einem projektunabhängigen Experten auf seine Eignung bezüglich verschiedenen Aspekten in einem Unternehmen hin in Abschnitt 8.2 bewertet.

Einige Kriterien der Problemstellung (siehe Kapitel 5) konnten in Experimenten evaluiert werden. Dies wird in Abschnitt 8.3 erläutert.

8.1 Industrie- und Praktikumsfallstudien

In diesem Abschnitt präsentieren wir die Ergebnisse und die Interpretation von zwei durchgeführten Fallstudien. Da der Aufbau der Studien gleich ist, stellen wir in diesem Abschnitt beide Studien gemeinsam vor.

Wir konnten den Modellierungsansatz einerseits in der Industrie einsetzen und andererseits in verschiedenen App-Entwicklungsprojekten im Rahmen von Praktika an der TU München. Wir sprechen in diesem Abschnitt von Industriestudie wenn wir die Studie im Rahmen der Anwendung des Modellierungsansatzes in der Industrie meinen und von Praktikumsstudie, wenn wir den Anwendung des Modellierungsansatzes im Rahmen der App-Entwicklungs-Praktika an der TU München meinen.

8.1.1 Kontext

Dieser Abschnitt führt in den Kontext der Industriestudie (Abschnitt 8.1.1.1) und der Praktikumsstudie (Abschnitt 8.1.1.2) ein.

8.1.1.1 Kontext Industriestudie

Das Softwareprojekt hatte einen Umfang von ca. 2 Personenjahren. Es wurde eine agile Vorgehensweise angewendet. In frühen Phasen des Requirements Engineerings wurde der Modellierungsansatz dieser Arbeit angewendet. Am Entwicklungsprozess waren insgesamt vier Personen beteiligt. Es handelte sich daher um ein kleines Entwicklungsprojekt.

Wir nehmen die Einteilung des Kontextes nach den Vorgaben von Wohlin et al. [WRH⁺12] vor.

Selbständig. Nach dem Einführungsworkshop wurden alle Ergebnisse in Form des Basismodells und den darauf aufbauenden Artefakten vom Entwicklungsteam erarbeitet.

Professionelle Softwareentwicklung. Die Fallstudie wurde in einem professionellen Softwareentwicklungsprojekt durchgeführt. Vom Entwicklungsteam wurde auch das Lastenheft erstellt. In frühen Phasen des Requirements Engineerings wurde der Modellierungsansatz, der in dieser Arbeit vorgestellt wird, verwendet. Nach der initialen Findung einer gemeinsamen Gesamtvorstellung wurde *Scrum* eingesetzt. Die Gesamtvorstellung diente auch als Basis für die Artefakte im Lastenheft. Das Ba-

sismodell diente in nachfolgenden Schritten im *Scrum*-Prozess als Basis für weitere Artefakte, die im Rahmen der Entwicklung der Individualsoftware erstellt wurden.

Reales Projektumfeld. Das entstandene Softwareprodukt wird inzwischen, obwohl es initial als Individualsoftware geplant wurde, in einigen Unternehmen eingesetzt. Das Projektumfeld entsprach dem eines realen Projektes, wie es im industriellen Umfeld zu finden ist.

Allgemeines Projekt. Das Projektumfeld war kein für das Projekt besonders zusammengestelltes Umfeld.

8.1.1.2 Kontext Praktikumsstudie

Wir nehmen die Einteilung des Kontextes nach den Vorgaben von Wohlin et al. [WRH⁺12] vor.

Selbständig. Nach dem Einführungsworkshop wurden alle Ergebnisse von den Probanden selbständig erarbeitet. Die Vorgehensweise wurde von den Probanden ausgeführt und das Basismodell erstellt. Das Basismodell wurde dann vom Rest des Teams weiter verwendet. Auch dies geschah selbständig.

Professionelle Softwareentwicklung im studentischen Praktikum. Die Fallstudie wurde im Rahmen der Entwicklung von Apps durchgeführt. Die Teams hatten Größen von drei bis fünf Personen. Alle Phasen der professionellen Softwareentwicklung wurden dabei durchlaufen: (a) Requirements Engineering, (b) Architektur-Entwurf, (c) Feinentwurf, (d) Implementierung und (e) Testen. Nach der initialen Requirements Engineering Phase wurde *Scrum* [Glo11] als Vorgehensmodell eingesetzt. Hier sei angemerkt, dass die Projekte nicht, wie in *Scrum* üblich, mit der Erstellung von UserStories begannen, sondern ein typischer Requirements-Engineering-Ansatz gewählt wurde, bestehend aus Erhebung, Analyse/Modellierung, Spezifikation und Validierung. Auf Basis der Artefakte wurden dann die Backlog-Items für den typischen *Scrum*-Prozess erstellt.

Reale Softwareprojekte. Die im Rahmen der Praktika erstellten Projekte sind Softwareprojekte, die in Teams entwickelt wurden und Anwendung in der Realität finden. Es handelt sich dabei nicht um Prototypen. Die Apps werden, wie es in einem professionellen Umfeld üblich ist, entwickelt. Der Aufwand der entwickelten Apps entspricht ca. 7 bis 10 Personenmonate (geschätzt auf Basis der durchgeführten Aufwandsschätzungen durch die Entwickler und ausgewertet von der Werkzeugunterstützung).

Allgemeines Projekt. Der angewendete Requirements-Engineering-Prozess ist unabhängig vom Projekt-Typ. Der Kontext ist daher initial allgemeingültig. Einschränkungen werden in Abschnitt 8.1.8 erläutert.

8.1.2 Forschungsziel

Wir verwenden zur Formulierung des Forschungszieles die Schablone von Wohlin et al. [WRH⁺12].

- ▶ **Analysiert wird:** Die Vorgehensweise des Modellierungsansatzes und Verwendung des Produktes als Basis für Artefakte, erstellt in frühen Phasen des Requirements Engineerings
- ▶ **Für den Zweck:** Der Evaluation
- ▶ **Im Bezug auf:** Die Anwendbarkeit und den Nutzen
- ▶ **Aus Sicht von:** Softwareentwicklern und Anforderungsingenieuren
- ▶ **Im Kontext:** Der frühen Phasen des Requirements Engineerings in einem Softwareentwicklungsprojekt

8.1.3 Forschungsfragen

Im Rahmen der Fallstudie soll Forschungsfrage 4 teilweise beantwortet werden.

RQ4: Inwiefern hilft der erarbeitete Ansatz bei der Lösung der Probleme bezüglich RQ3 bei der Erhebung von Problem- und Lösungsdomäne in frühen Phasen des Requirements Engineerings?

Um die Forschungsfrage beantworten zu können, müssen die einzelnen Kriterien die im Rahmen der Beantwortung von Forschungsfrage 3 gefunden wurden, adressiert werden.

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

Die Forschungsfrage 3 wird in Kapitel 5 beantwortet. Nicht alle in Kapitel 5 erarbeiteten Kriterien, konnten im Rahmen der Studie evaluiert werden.

Im Rahmen der Fallstudie wird ein Beitrag zur Evaluierung folgender Kriterien¹ geleistet:

- ▶ **K1.1: Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.**

Im Rahmen der Durchführung der Fallstudie wurden die Entwicklungsteams gebeten zu bewerten, ob inhaltliche Missverständnisse durch die Anwendung des Ansatzes in frühen Phasen geklärt werden konnten. Hierzu leisten wir mit der Fallstudie einen Beitrag.

Die Interpretation eines Basismodells von verschiedenen Probanden wurde im Experiment in Abschnitt 8.3 untersucht.

- ▶ **K2.1: Die Artefakte stehen in einer Beziehung zueinander, sodass keine Interpretationslücken geschlossen werden müssen.**

¹ Die Kriterien stammen aus Kapitel 5.

Im Rahmen der Fallstudie wurden Artefakte auf Basis des Basismodells erstellt. Es wurde im Rahmen der Fallstudie darauf geachtet, dass die erstellten Artefakte einen Bezug zum Basismodell haben und dadurch die Artefakte auf einer ähnlichen Abstraktionsebene sind. Im Anschluss wurden verschiedene Aspekte dieses Kriteriums bewertet.

- ▶ **K3.1: Die Artefakte, erstellt in frühen Phasen des Requirements Engineerings, stehen in keinem Widerspruch untereinander.**

Die im Rahmen der Fallstudie erstellten Artefakte sollen in keinem Widerspruch zueinander stehen. Nach Anwendung des Modellierungsansatzes bewerteten die Entwicklungsteams die Konsistenz der darauf basierenden Artefakte untereinander.

- ▶ **K4.1: Die Artefakte, die gleiche Ausschnitte eines Problems bzw. einer Lösung mit Hilfe der gleichen Modellierungstechnik beschreiben, werden von unterschiedlichen Personen ähnlich modelliert.**

Im Rahmen der Praktikumsstudie können wir die erstellten Artefakte dahingehend untersuchen, ob die Verknüpfung zu gleichen Artefakt-Typen aus den verschiedenen Basismodellen der Projekte zu ähnlichen Inhalten in den Artefakten führen.

- ▶ **K5.1: Alle Ausschnitte eines Gesamtzusammenhangs werden von zumindest einem Artefakt beschrieben.**

Die Entwicklungsteams bewerteten nach Anwendung des Modellierungsansatzes, ob alle im Basismodell vorhandenen Sacherverhalte von weiteren Artefakten beschrieben worden sind.

Alle Artefakte, die im Rahmen der Fallstudien erstellt wurden, wurden auf Basis des Basismodells erstellt. Sie beschreiben einen Ausschnitt aus dem Basismodell. Das Basismodell repräsentiert einen Gesamtzusammenhang. Dieser wurde nach Verwendung als Basis für weitere Artefakte bewertet.

Abschließend wurde bewertet, ob das Wissen, in den auf das Basismodell basierenden Artefakten, ausgeglichen² repräsentiert ist.

- ▶ **K6.1: Eine Verknüpfung zu anderen Artefakten ist angemessen möglich.**

Im Rahmen der Überprüfung der Erfüllung dieses Kriteriums wurden die Modellierer gebeten, anhand der auf dem Basismodell basierenden Artefakte zu bewerten, ob die Verknüpfung zu anderen Modellen angemessen möglich war.

- ▶ **K7.1: Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.**

Im Rahmen der Überprüfung der Erfüllung dieses Kriteriums wurden die Probanden gebeten zu bewerten, ob der Modellierungsansatz dabei hilft, die Kommunikation zwischen den Stakeholdern zu verbessern und in wie weit das Basismodell dazu geeignet ist, ein Gesamtverständnis zu repräsentieren.

Wir sind nicht in der Lage, die Erfüllung aller Kriterien durch die Fallstudien zu zei-

² Wir sprechen in diesem Zusammenhang von einer ausgeglichenen Repräsentation, wenn jede Entität und stereotypisierte Beziehung im Basismodell von zumindest einem weiteren Artefakt beschrieben wird.

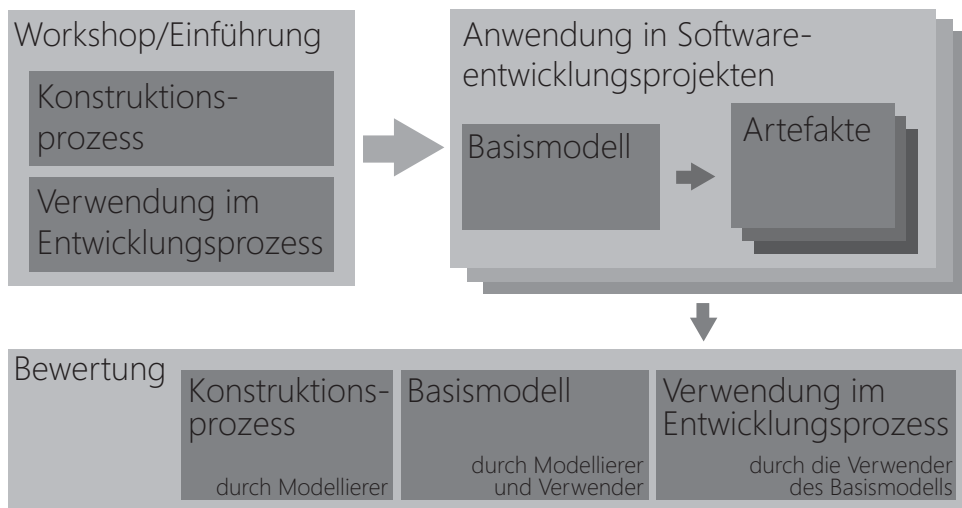


Abbildung 8.2: Vorgehen zur Datenerhebung

gen. Für die Beantwortung der offenen Fragen wurde ein Laborexperiment durchgeführt, siehe dazu Abschnitt 8.3.

Zusätzlich haben wir im Aufbau der Studie und des darin vorgestellten Fragebogens auch Fragen berücksichtigt, die auf die Evaluierung der Anwendbarkeit des Ansatzes abzielen.

8.1.4 Studien-Aufbau und Erhebung

Abbildung 8.2 zeigt unsere Vorgehensweise. Anfänglich wurden alle Personen im Ansatz geschult. Für die einzelnen Projekte wurde die Rolle des Modellierers festgelegt. Der Modellierer ist für das Erstellen des Basismodells verantwortlich. Dazu wendet der Modellierer die vorgeschlagene Vorgehensweise an. Im Anschluss an die Anwendung bewertet dieser die Vorgehensweise sowie die Aspekte des Basismodells, die für das Finden einer gemeinsamen Vorstellung relevant sind.

Das erstellte Basismodell dient nun in weiteren Schritten des Requirements Engineerings als Basis für das Erstellen von verschiedenen Artefakten. Diese Weiterverwendung und der Nutzen für den Entwicklungsprozess wird vom gesamten Entwicklungsteam bewertet.

Bewertung. Die Bewertung wurde mit Hilfe von Fragebögen durchgeführt. Die Personen, die die Rolle des Modellierers inne hatte, bewerteten die Vorgehensweise zur Erstellung des Basismodells und dessen Eigenschaften unabhängig vom Rest des Entwicklerteams. Die restlichen Entwicklerteams bewerteten das Basismodell und dessen Verwendung im Entwicklungsprozess separat. Jedes Entwicklerteam musste sich zur Bewertung eine gemeinsame Meinung über die Verwendung des Basismodells bilden. Die Bewertung entspricht daher der Team-Bewertung. Wir haben uns in diesem Schritt für eine Team-Bewertung entschieden, da dies innerhalb der Entwicklerteams zu weiteren Diskussionen über den Modellierungsansatz und die zu bewertenden Facetten geführt hat.

Die Fragebögen wurden nach Kriterien organisiert, die wir verwendeten, um die Forschungsfrage 4 zu beantworten. Für die einzelnen Kriterien wurden offene und geschlossene Fragen gestellt. Die geschlossenen Fragen boten Antwortmöglichkeiten in einer achtstufigen Skala von *trifft völlig zu* bis *trifft überhaupt nicht zu*. Um eine neutrale Aussage zu vermeiden, wurde eine achtstufige Skala verwendet.

Die Fragebögen boten nach jeder Frage die Möglichkeit freie Anmerkungen zu machen (davon wurde häufig Gebrauch gemacht).

Fragebogen-Design, Bewertung durch Modellierer des Basismodells. Nachstehend die geschlossenen Fragen (GF) zur Bewertung des Vorgehens durch die Modellierer (M):

M-GF1 Die Vorgehensweise ist einfach anzuwenden: Die einzelnen Schritte der Vorgehensweise sind einfach anzuwenden.

M-GF2 Die Vorgehensweise ist strukturiert: Die Vorgehensweise gibt strukturiert vor, wie vorgegangen werden soll.

M-GF3 Das Vorgehen ist eindeutig: Es ist zu jedem Zeitpunkt klar, welche Entitäten zu erfassen sind.

M-GF4 Die Konzepte sind eindeutig: Die Bedeutung der auszuwählenden Stereotypen ist eindeutig klar.

M-GF5 Die Produktivität wird gefördert: Der Ansatz unterstützt das Requirements Engineering insofern, dass er die Produktivität erhöht – z. B. verbessert er die Identifikation von möglichen Use-Cases, Datenmodellen, usw.

M-GF6 Das Vorgehen ist nicht fehleranfällig: Das Vorgehen ist klar genug spezifiziert, dass Fehler in der Anwendung des Vorgehens nicht einfach auftreten können.

M-GF7 Das Basismodell fördert die Kommunikation unter den involvierten Stakeholdern.

M-GF8 Das Vorgehen eignet sich, um in frühen Phasen des Requirements Engineerings eingesetzt zu werden.

M-GF9 Vergleich zur bisherigen Vorgehensweise: Ist der Ansatz im Vergleich zu ihrer bisherigen Vorgehensweise besser geeignet, um in frühen Phasen des Requirements Engineerings eine frühe Vorstellung des Problem- und Lösungsdomäne zu erarbeiten?

Fragebogen-Design, Bewertung durch Verwender des Basismodells. Nachstehend werden die geschlossenen Fragen (GF) aufgelistet, die dazu dienen, die Verwendung (V) des Basismodells im Requirements Engineering in frühen Phasen zu bewerten.

V-GF1 Das Basismodell ist vollständig: Alle nach Ihrer Einschätzung relevanten Aspekte sind nach der Erstellung im Basismodell enthalten.

V-GF2 Das Basismodell ist zur Identifikation geeignet: Use-Cases, Datenmodelle usw. konnten auf Basis des Basismodells geeignet identifiziert werden.

- V-GF3 Die Abstraktionsebene des Basismodells ist für ihren Zweck angebracht.
- V-GF4 Die Inhalte des Basismodells sind so eindeutig, dass sie für ihren Zweck weiter verwendet werden können.
- V-GF5 Das Basismodell hilft, inhaltliche Missverständnisse innerhalb der Stakeholder zu klären.
- V-GF6 Die Abhängigkeiten zwischen Artefakten, die auf dem Basismodell basieren, werden klar.
- V-GF7 Das Basismodell als Basis führt zu besser strukturierten Artefakten.
- V-GF8 Das Basismodell repräsentiert die Vorstellung über das Gesamtsystem so, dass diese klar ist und angebracht repräsentiert wird.
- V-GF9 Das Basismodell ist minimalistisch: Das Basismodell enthält nur die nötigsten Entitäten.
- V-GF10 Das Basismodell fördert die Kommunikation unter den involvierten Stakeholdern.
- V-GF11 Das Basismodell eignet sich für den Einsatz in frühen Phasen des Requirements Engineerings.
- V-GF12 Das Vorgehen, das Basismodell als Basis für weitere Artefakte in frühen Phasen des Requirements Engineerings zu verwenden, eignet sich besser, als die zuvor verwendete Vorgehensweise.
- V-GF13 Die Verwendung des Basismodells als Basis für weitere Artefakte führt zu qualitativ besseren Artefakten.
- V-GF14 Die Verwendung des Basismodells als Basis für weitere Artefakte führt zu konsistenten Artefakten.
- V-GF15 Das Basismodell hilft, das Wissen über Problem- und Lösungsdomäne ausgeglichen in Artefakten zu erheben: Ausgeglichen ist das Wissen in Artefakten dann, wenn alle Ausschnitte des Basismodells in anderen Artefakten detailliert wurden

Industriestudie Fall- und Subjektauswahl. Um den in dieser Arbeit vorgeschlagenen Modellierungsansatz evaluieren zu können, muss dieser in einem Softwareentwicklungsprojekt integriert werden. Gefunden werden musste daher ein Entwicklungsteam, das bereit war, den Modellierungsansatz anzuwenden und diesen nach der Anwendung zu bewerten, sowie möglichst viele der erstellten Artefakte zur Analyse zur Verfügung zu stellen. Im Rahmen der Anwendung des Modellierungsansatzes wurden zwei Rollen benannt, die an der Ausführung des Modellierungsansatzes beteiligt sind: (a) der Modellierer, der für die Erstellung des Basismodells verantwortlich ist und dafür Sorge tragen muss, dass genügend Personen in den Prozess involviert sind und (b) die Verwender, die das Basismodell als Basis für weitere Artefakte im Softwareentwicklungsprozess verwenden.

Praktikumsstudie Fall- und Subjektauswahl. Um den in dieser Arbeit vorgeschlagenen Modellierungsansatz evaluieren zu können, muss dieser in einem Soft-

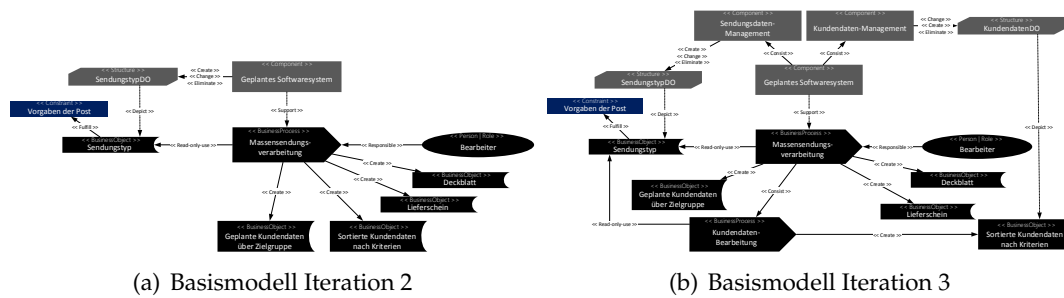


Abbildung 8.3: Basismodelle nach Iteration 2 und 3

warentwicklungsprojekt integriert werden. Zusätzlich muss gegeben sein, dass die erstellten Artefakte analysierbar sind (daher zugänglich) und im Rahmen der Arbeit auch verwendet werden können. Häufig ist dies in Softwareentwicklungsprojekten in der Industrie nicht gegeben, wir haben uns daher entschlossen, zusätzlich zur durchgeführten und zugänglichen Fallstudie in der Industrie, den Ansatz im Rahmen der Entwicklung von Apps für Tablets und Smartphones in einem Praktikum an der TU München zu erproben.

Analyse-Methode. Wegen des gewählten Studien-Designs ist statistisches Testen von Hypothesen nicht möglich. Wir präsentieren in den Ergebnissen die erstellten Basismodelle sowie die Antworten auf die geschlossenen Fragen des Fragebogens.

Ablauf. Nach einem Einführungsworkshop wird der Ansatz von den Entwicklerteams angewendet. Im Rahmen der Anwendung sollte ein Basismodell und darauf aufbauende Artefakte erstellt werden. Abschließend soll nach der Requirements-Engineering-Phase eine Bewertung vorgenommen werden, wie sie in diesem Abschnitt beschrieben wurde.

8.1.5 Ergebnisse Industriestudie

Der Modellierer erstellte mit den Domänenexperten das Basismodell. Die Rolle des Softwareentwicklers wurde vom Modellierer eingenommen. Die Domänenexperten waren die Auftraggeber der Individualsoftware. Am Erstellungsprozess waren drei Domänenexperten beteiligt. Nach der Erstellung des Basismodells wurde dieses verwendet, um eine weitere Detaillierung von Ausschnitten vorzunehmen. Es wurde bei der Verwendung darauf geachtet, die auf dem Basismodell aufbauenden Artefakte stets konsistent zum Basismodell zu halten. Einige Detailfragen mussten im Rahmen der Erstellung der weiteren Artefakte von den Domänenexperten erfragt werden. Die Erstellung der weiteren Artefakte geschah in Abwesenheit der Domänenexperten.

Abbildung 8.3 zeigt die einzelnen Schritte beim Erstellen des Basismodells. Iteration 1 wurde hier weggelassen, da sie keine zusätzlichen Informationen beinhaltet.

Abbildung 8.4 zeigt das Basismodell nach der Iteration 4.

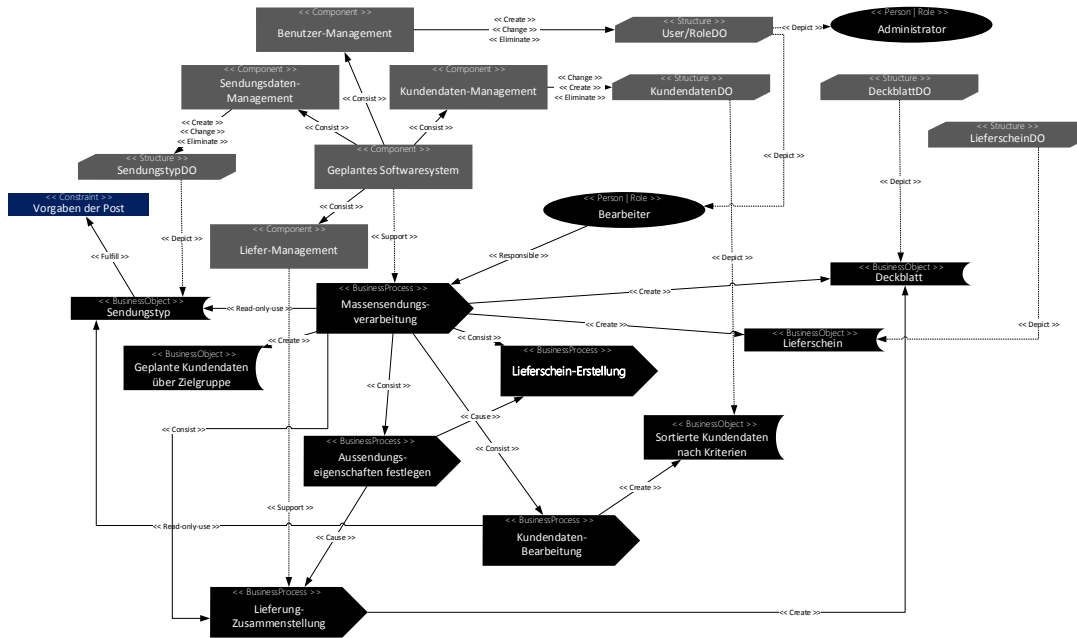
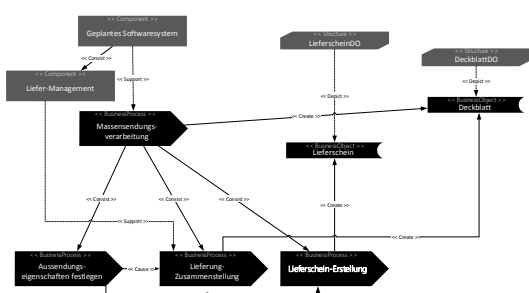
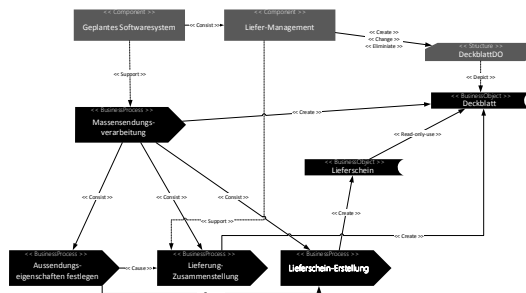


Abbildung 8.4: Basismodell nach Iteration 4



(a) Basismodell nach Iteration 4, Ausschnitt



(b) Basismodell nach Iteration 5, Detaillierung eines Ausschnittes

Abbildung 8.5: Ausschnitt aus dem Basismodell aus Iteration 4 in Teilbereich (a) und nach Iteration 5 in Teilbereich (b)

Abbildung 8.5 zeigt in Teil (a) einen Ausschnitt aus dem Basismodell in Abbildung 8.4, der in Iteration 5, abgebildet in Teil (b) weiter detailliert wurde.

Abbildung 8.6 zeigt die Bewertung des Modellierungsansatzes durch die Rolle des Modellierers.

	trifft völlig zu				trifft überhaupt nicht zu				
M-GF1		X							Vorgehen ist einfach
M-GF2	X								Vorgehen ist strukturiert
M-GF3		X							Vorgehen ist eindeutig
M-GF4		X							Konzepte sind eindeutig
M-GF5		X							Ansatz fördert Produktivität
M-GF6			X						Ansatz ist nicht fehleranfällig
M-GF7	X								Ansatz fördert Kommunikation
M-GF8	X								Eignet sich in frühen Phasen des RE
M-GF9	X								Besser, als bisherige Vorgehensweise

Abbildung 8.6: Bewertungen durch die Rolle des Modellierers

Abbildung 8.7 zeigt die Bewertung des Basismodells und des Modellierungsansatzes durch das Team. Bewertet wurde die Verwendung des Basismodells.

Bemerkungen.

- I-K1 „Werkzeug-Unterstützung wäre notwendig, die Entitäten sind schwer zu verwalten.“
- I-K2 „Der Sinn der Abbildung von Geschäftsobjekten in Strukturen muss besser erläutert werden, das ist anfänglich verwirrend.“
- I-K3 „Wir haben bei der Erstellung des Modells sicher bereits früher intensiv diskutiert, als wir es vorher in Projekten gemacht haben.“
- I-K4 „Die Idee ein Leitmodell im Requirements Engineering zu verwenden ist sehr hilfreich.“

8.1.5.1 Verknüpfung zu anderen Modellen und entstandenes Produkt

Auf Basis des Basismodells wurden weitere Artefakte des Projektes erstellt. Das Basismodell wurde dabei konsistent gehalten.

Abbildung 8.8 zeigt einen Ausschnitt aus den entstandene Artefakten während des Feinentwurfes. Die im Basismodell initial in deutsch erfassten Komponenten wurden im Feinentwurf in Englisch erfasst, wie in Abbildung 8.8.

	trifft völlig zu		trifft überhaupt nicht zu		
V-GF1	X				Basismodell ist vollständig
V-GF2	X				Zur Identifikation geeignet
V-GF3	X				Abstraktionsebene ist angebracht
V-GF4	X				Inhalte des Basismodells sind eindeutig
V-GF5	X				Basismodell klärt inhaltl. Missverständnisse
V-GF6	X				Klärung von Abhängigkeiten zwischen Artefakten
V-GF7		X			Besser strukturierte Artefakte durch Basismodell
V-GF8	X				Gesamtsystemvorstellung ist klar
V-GF9			X		Basismodell ist minimalistisch
V-GF10	X				Basismodell fördert Kommunikation
V-GF11	X				Basismodell eignet sich für frühe Phasen des RE
V-GF12	X				Besser als vorher verwendeter Ansatz
V-GF13	X				Qualität von Artefakten verbessert sich
V-GF14	X				Konsistenz von Artefakten verbessert sich
V-GF15	X				Ausgeglichen verteiltes Wissen in Artefakten

Abbildung 8.7: Bewertungen durch die Verwender des Basismodells, dem Team

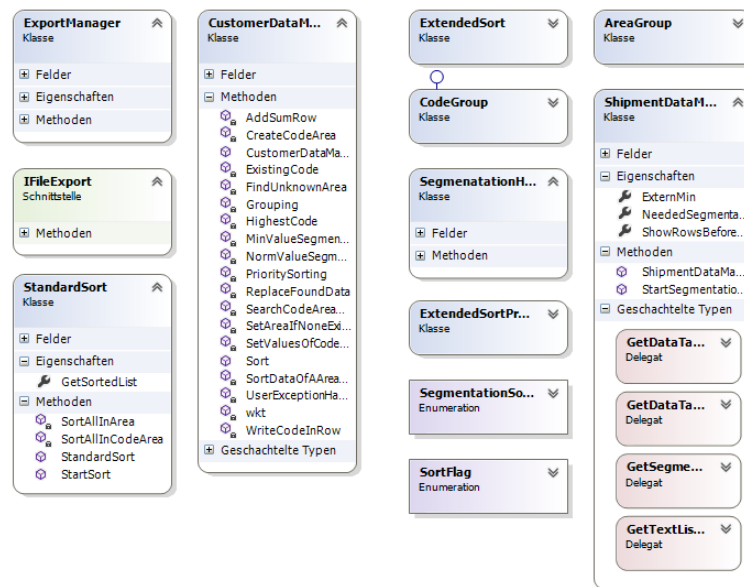
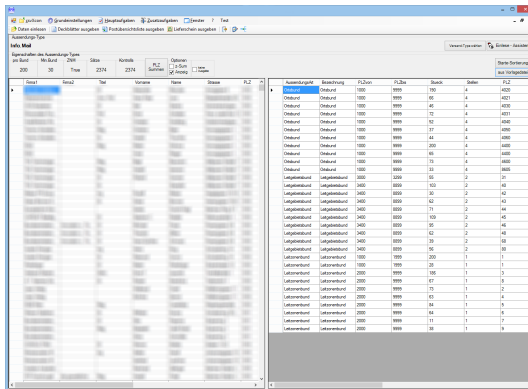
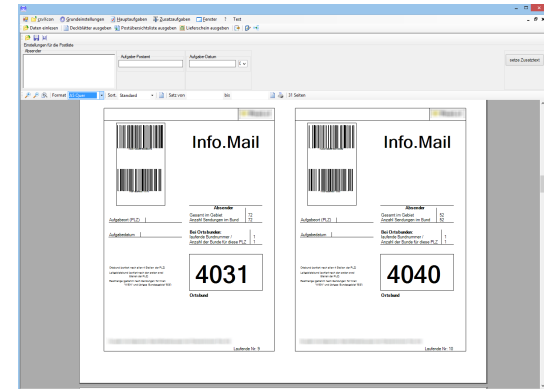


Abbildung 8.8: Ausschnitt aus Artefakten des Feinentwurfs, u.a. zu sehen sind das Kunden-
daten-Management als *CustomerDataManagement* und das Sendungsdaten-
Management als *ShipmentDataManagement* aus dem Basismodell

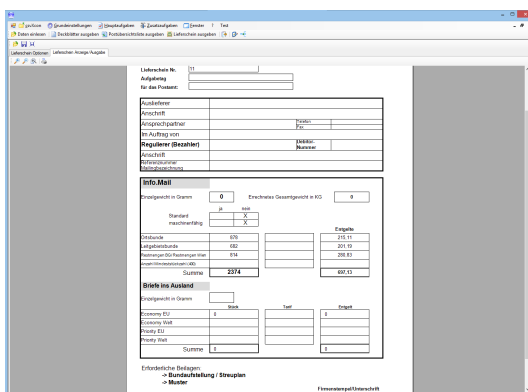
In Abbildung 8.8 ist das im Basismodell erfasste Kundendaten- und Sendungsdaten-Management erfasst. Diese im Basismodell als Komponenten erfassten Entitäten wurden im Feinentwurf, und damit auch in der Implementierung, als Klasse repräsentiert.



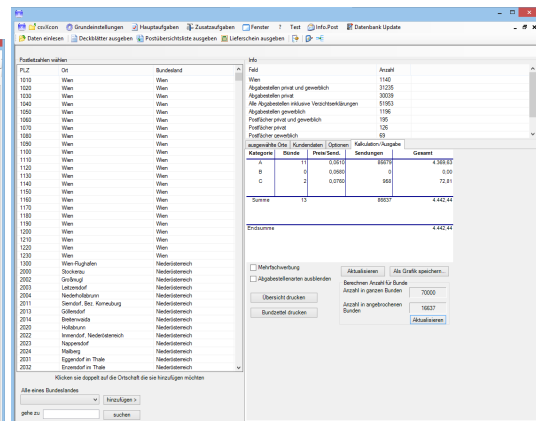
(a) Geschäftsprozess *Massensendungsverarbeitung*, initiale Ansicht



(b) Repräsentation der Struktur *Deckblatt*



(c) Geschäftsprozess *Lieferschein-Erstellung*



(d) Geschäftsprozess *Aussendungseigenschaften festlegen*

Abbildung 8.9: Screenshots aus der entstandenen Anwendung

Abbildung 8.9 zeigt Screenshots der entstandenen Anwendung im Prototypen-Stadium. In Teil (a) und (b) mussten wir Inhalte verdecken.

8.1.6 Ergebnisse Praktikumsstudie

Dieser Abschnitt zeigt die Ergebnisse der Praktikumsstudie.

8.1.6.1 Ergebnisse Erstellung des Basismodells

Kommentare der Modellierer. Nachstehend haben wir die von den Modellierern (M) getätigten Aussagen im Rahmen der Bewertung aufgelistet, wobei jeder Eintrag einem Zitat entspricht:

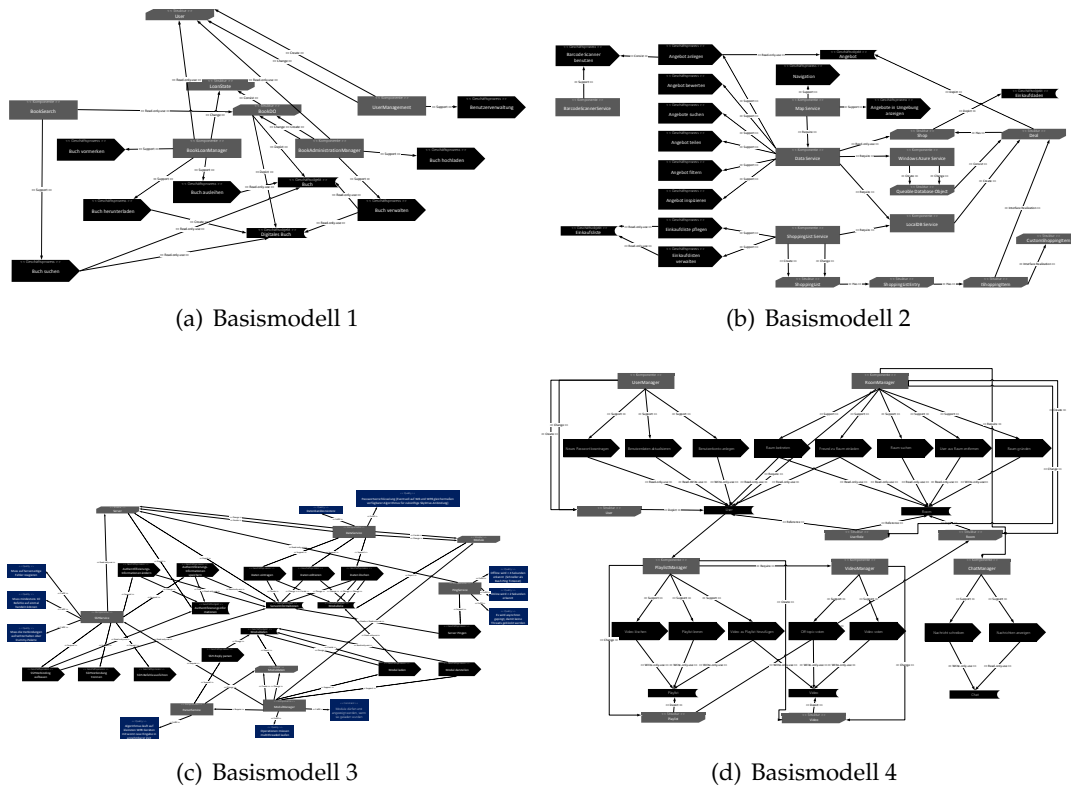


Abbildung 8.10: Ausschnitt aus der im Rahmen der Fallstudie erstellten Basismodelle

	trifft völlig zu				trifft überhaupt nicht zu				
M-GF1	2	1	2	1	0	0	0	0	Vorgehen ist einfach
M-GF2	3	3	0	0	0	0	0	0	Vorgehen ist strukturiert
M-GF3	3	3	0	0	0	0	0	0	Vorgehen ist eindeutig
M-GF4	2	2	2	0	0	0	0	0	Konzepte sind eindeutig
M-GF5	3	2	1	0	0	0	0	0	Ansatz fördert Produktivität
M-GF6	1	3	2	0	0	0	0	0	Ansatz ist nicht fehleranfällig
M-GF7	6	0	0	0	0	0	0	0	Ansatz fördert Kommunikation
M-GF8	3	3	0	0	0	0	0	0	Eignet sich in frühen Phasen des RE
M-GF9	2	3	1	0	0	0	0	0	Besser, wie bisherige Vorgehensweise

Abbildung 8.11: Bewertungen, Anzahl von Stimmen zu gekürzten Aussagen durch Rolle Modellierer

- M-K1 „Mir gefällt sehr, dass klar zwischen den Entitäten in der Problemwelt und der Lösungswelt differenziert wird. Auch wenn ich mich überwinden musste die Entitäten auf den ersten Blick redundant zu modellieren, so macht diese Trennung durchaus Sinn.“
- M-K2 „Sehr gut gefällt mir, dass mit einem solchen Basismodell mehrere herkömmliche Modelle gleichzeitig abgedeckt werden. Dadurch hatte ich nach dem Erstellen des Basismodells bereits eine genaue Vorstellung meiner Use-Cases aber auch meiner auftretenden Datenobjekte.“
- M-K3 „Gute Modellierung der realen Welt in einem technischen Kontext. Gute Kommunikation innerhalb des Teams mit Übergang vom RE zur technischen Denkweise.“
- M-K4 „Genaue Abgrenzung der realen von der digitalen Domäne.“
- M-K5 „Eindeutiges Vorgehen. Mit der Vorlage nicht schwer umzusetzen.“
- M-K6 „Das Basismodell stellt definitiv einen Mehrwert im Bezug zur bisherigen UML-Modellierung dar.“
- M-K7 „Die Modellierung der Problemwelt und Lösungswelt führt zu sehr vielen Beziehungen, die auf den ersten Blick ähnliche Aussagen treffen.“
- M-K8 „Man braucht eine gewissen Einarbeitungszeit.“

8.1.6.2 Ergebnisse Verwendung des Basismodells

Abbildung 8.12 zeigt verschiedene aus den Basismodellen, die im Rahmen der Projekte erstellt wurden, erstellte Modelle in anderen Modellierungstechniken.

Kommentare der Teams. Nachstehend haben wir die von den Teams (T) getätigten Aussagen im Rahmen der Bewertung aufgelistet, wobei jeder Eintrag einem Zitat entspricht:

- T-K1 „Guter Überblick und leichter Ansatz.“
- T-K2 „Führt zu gutem Rich-Picture, zur besseren Repräsentation des Umfelds.“
- T-K3 „Das Modell stellt sehr gut die Verbindung zwischen Artefakten her.“
- T-K4 „Bessere Beschreibung der Entitäten und Beziehungen erwünscht.“
- T-K5 „Über die innere Struktur von Software-Komponenten wird nicht viel erfasst.“

8.1.7 Analyse

Wir analysieren die erhobenen Daten anhand ihres Beitrages zur Erfüllung der Kriterien.

Eine zusammenfassende Diskussion der Ergebnisse findet in Abschnitt 8.4 statt.

Die Bewertung, wie im Aufbau in Abschnitt 8.1.4 erwähnt, erfolgte in einer achtstufigen Skala, von *Trifft völlig zu* bis *Trifft überhaupt nicht zu*. Wir sprechen im Folgenden von acht Punkten, wenn die Bewertung *Trifft völlig zu* gewählt wurde und von einem

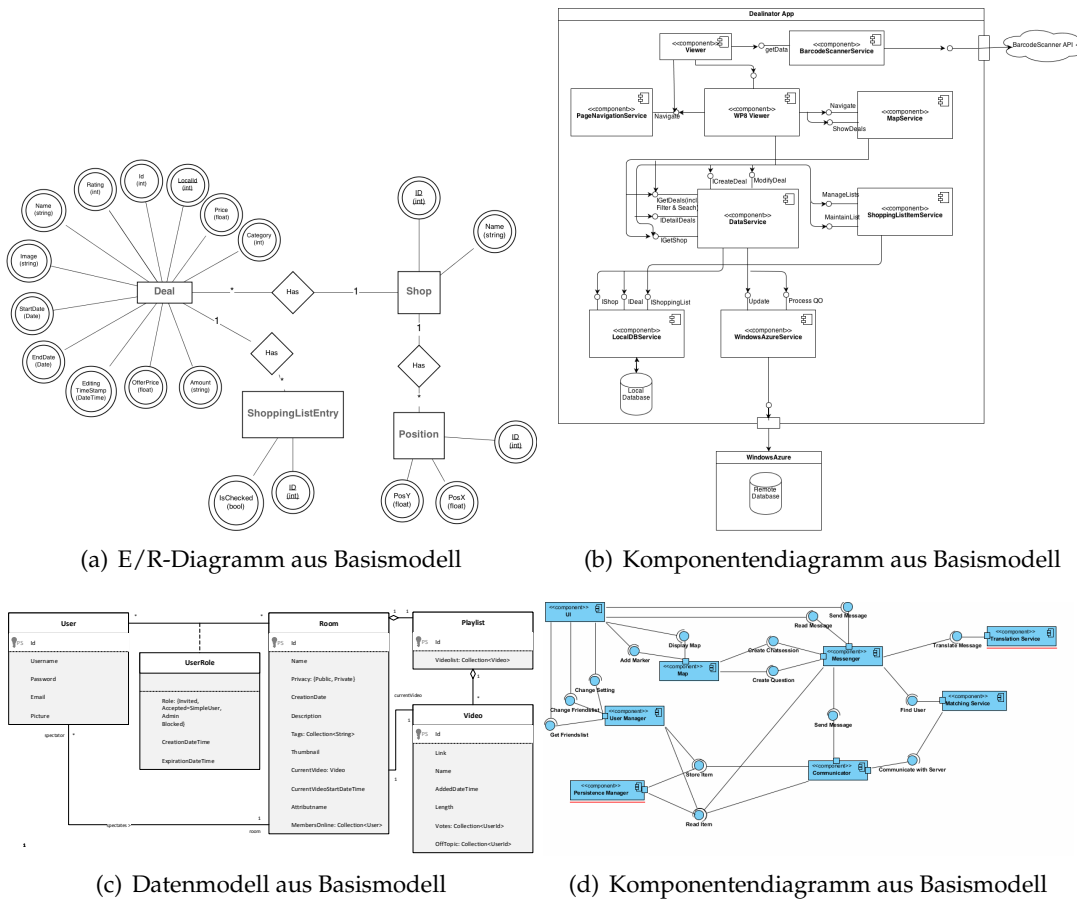


Abbildung 8.12: Ausschnitt aus Artefakten, erstellt auf Basis der Basismodelle

Punkt, wenn *Trifft überhaupt nicht zu* gewählt wurde. Die achtstufige Skala erlaubte es uns neutrale Aussagen zu vermeiden. Wir sprechen daher ab 5 Punkten und mehr, von einer eher zutreffenden Aussage und bei 4 Punkten und weniger, von einer eher nicht zutreffenden.

► **K1.1: Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.**

Wir analysieren den Einfluss auf die Klärung von inhaltlichen Missverständnisses durch die Verwendung des Modellierungsansatzes.

Industriestudie. Die Verwender des Basismodells bewerteten, ob das Basismodell die Kommunikation fördert (Frage V-GF10), dies wurde mit acht Punkten bestätigt. Zusätzlich wurde bewertet, ob das Basismodell zur Identifikation von Inhalten für darauf aufbauende Artefakte (z. B. Use-Cases, usw.) geeignet ist (Frage V-GF2). Dies wurde mit acht Punkten bestätigt. Missverständnisse lassen sich dann reduzieren, wenn die damit erfassten Inhalte so eindeutig wie möglich, aber für ihre Phase im Softwareentwicklungsprozess angebracht, sind. Dies wurde in Frage V-GF4 erhoben, es wurden sieben Punkte gegeben.

Zusätzlich wurde in Frage V-GF5 gefragt, ob das Basismodell inhaltliche Miss-

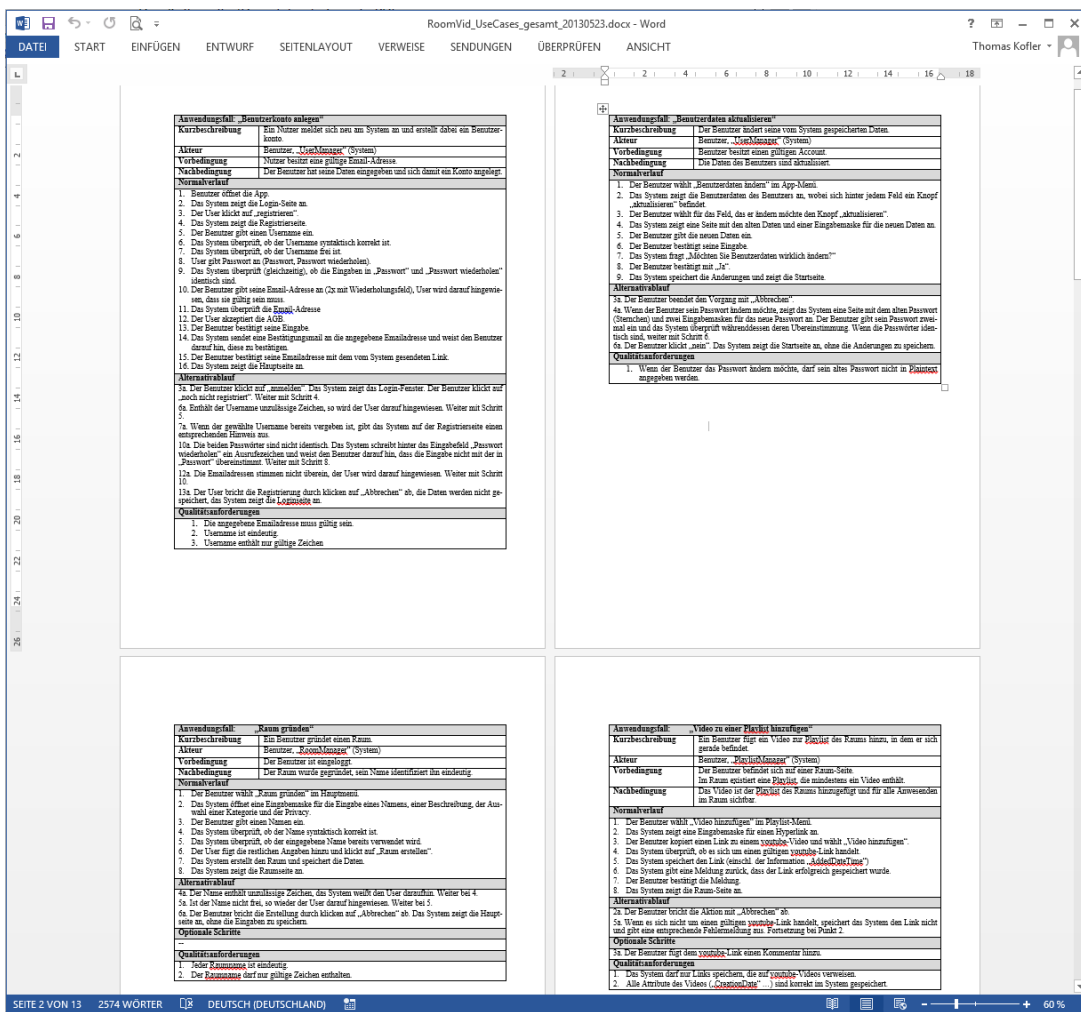


Abbildung 8.13: Ausschnitt aus den im Rahmen eines Projektes erstellten Use-Cases die auf dem Basismodell basieren

verständnisse klärt. Dies wurde mit acht Punkten bestätigt.

Der Modellierer des Basismodells hat unabhängig von den Verwendern bewertet, ob die Kommunikation bei der Erstellung des Basismodells gefördert wird (Frage M-GF7). Dies wurde mit sieben Punkten bewertet.

Praktikumsstudie. Inhaltliche Missverständnisse können auftreten, wenn Modelle nicht vollständig (siehe dazu Definition in Abschnitt 6.4) sind. Die Bewertung des Basismodells auf Vollständigkeit erfolgte in Frage V-GF1 und wurde mit durchschnittlich 7,2 Punkten bewertet.

Die repräsentierten Inhalte in einem Basismodell müssen für die Stakeholder eindeutig sein. Dies wurde von den Verwendern in Frage V-GF4 mit durchschnittlich 7,2 Punkten bewertet.

Es wurde auch direkt die Frage nach der Klärung von inhaltlichen Missverständnissen durch das Basismodell in Frage V-GF5 gestellt. Dies wurde mit 7,5 Punkten bewertet.

	trifft völlig zu				trifft überhaupt nicht zu				
V-GF1	2	3	1	0	0	0	0	0	Basismodell ist vollständig
V-GF2	5	0	1	0	0	0	0	0	Zur Identifikation geeignet
V-GF3	4	2	0	0	0	0	0	0	Abstraktionsebene ist angebracht
V-GF4	2	3	1	0	0	0	0	0	Inhalte d. Basismodells sind eindeutig
V-GF5	4	1	1	0	0	0	0	0	Basismodell klärt inhaltl. Missverständnisse
V-GF6	4	2	0	0	0	0	0	0	Klärung von Abhängigkeiten zwischen Artefakten
V-GF7	5	0	1	0	0	0	0	0	Besser strukturierte Artefakte durch Basismodell
V-GF8	3	2	1	0	0	0	0	0	Gesamtsystemvorstellung ist klar
V-GF9	3	1	1	1	0	0	0	0	Basismodell ist minimalistisch
V-GF10	4	2	0	0	0	0	0	0	Basismodell fördert Kommunikation
V-GF11	5	1	0	0	0	0	0	0	Basismodell eignet sich für frühe Phasen des RE
V-GF12	6	0	0	0	0	0	0	0	Besser als vorher verwendeter Ansatz
V-GF13	3	3	0	0	0	0	0	0	Qualität von Artefakten verbessert sich
V-GF14	4	2	0	0	0	0	0	0	Konsistenz von Artefakten verbessert sich
V-GF15	3	3	0	0	0	0	0	0	Ausgeglichen verteiltes Wissen in Artefakten

Abbildung 8.14: Bewertungen, Anzahl von Stimmen zu gekürzten Aussagen, durch die Verwender (Teams) der Basismodelle

Die Interpretation des Basismodells wird in Abschnitt 8.3 genauer beleuchtet.

- **K2.1: Die Artefakte stehen in einer Beziehung zueinander, sodass keine Interpretationslücken geschlossen werden müssen.**

Industriestudie. In Frage V-GF6 haben wir erhoben, ob das Basismodell zur Klärung von Abhängigkeiten zwischen Artefakten geeignet ist, dies wurde mit 8 Punkten bestätigt.

In Frage V-GF2 wurde die Eignung des Basismodells als Basis für die Identifikation darauf aufbauender Artefakte bewertet, dies wurde mit acht Punkten bestätigt.

Praktikumsstudie. Zu einem ähnlichen Ergebnis führte die Bewertung durch Frage V-GF6 auch bei den Probanden der Praktikumsstudie, die dies mit durchschnittlich 7,7 Punkten bestätigten.

Die Eignung des Basismodells zur Identifikation der darauf aufbauenden Artefakte, wurde mit durchschnittlich 7,7 Punkte bewertet.

- **K3.1: Die Artefakte, erstellt in frühen Phasen des Requirements Engineerings, stehen in keinem Widerspruch untereinander.**

Industriestudie. Die Verwender des Basismodells bewerteten, ob die Verwendung des Basismodells zu konsistenten Artefakten führt, dies wurde in Frage V-GF14 mit sieben Punkten bestätigt.

Praktikumsstudie. Ähnlich der Industriestudie haben die Probanden der Praktikumsstudie diese Frage (V-GF14) mit durchschnittlich 7,7 Punkten bewertet.

- ▶ **K4.1: Die Artefakte, die gleiche Ausschnitte eines Problems bzw. einer Lösung mit Hilfe der gleichen Modellierungstechnik beschreiben, werden von unterschiedlichen Personen ähnlich modelliert.**

Weitere Aspekte dieses Kriteriums betrachten wir im Rahmen des Laborexperimentes in Abschnitt 8.3.

Wir haben im Rahmen der Evaluierung einige Fragen zur Modellbildung gestellt, da wir im Modellierungsansatz den Modellbildungsvorgang mit der Vorgehensweise steuern.

Industriestudie. Es wurde die Einfachheit des Vorgehens (Frage M-GF1) mit sieben Punkten, die Strukturiertheit des Vorgehens (Frage M-GF2) mit acht Punkten und die Eindeutigkeit des Vorgehens (Frage M-GF3) mit sieben Punkten bewertet.

Praktikumsstudie. Hierzu wurde erhoben, ob das Vorgehen einfach (Frage M-GF1), strukturiert (Frage M-GF2) und eindeutig (Frage M-GF3) ist, dies wurde mit durchschnittlich 6,7, 7,5 und 7,5 Punkten bewertet.

In den Artefakten, die im Rahmen des Praktikums von den unterschiedlichen Entwicklerteams erstellt wurden, haben wir uns u.a. die Datenmodelle angesehen. Wir haben festgestellt, dass z. B. die frühe Verwendung von Strukturen im Basismodell dazu führt, dass die Strukturen 1:1 als Entitäten in Datenmodellen verwendet werden. Dies ist über alle Entwicklerteams hinweg geschehen.

- ▶ **K5.1: Alle Ausschnitte eines Gesamtzusammenhangs werden von zumindest einem Artefakt beschrieben.**

Industriestudie. Die Verwender bewerteten, ob die auf dem Basismodell basierten Artefakte, Wissen ausgeglichen repräsentieren (Frage V-GF15), dies wurde mit sieben Punkten bewertet.

Praktikumsstudie. Die ausgeglichene Repräsentation von Wissen (Frage V-GF15) in den auf dem Basismodell basierenden Artefakten wurde mit durchschnittlich 7,5 Punkten bewertet.

Die Ausgeglichenheit spiegelt sich auch in der Struktur der auf dem Basismodell basierenden Artefakte wieder, dies wurde in Frage V-GF7 mit durchschnittlich 7,7 Punkten bestätigt.

- ▶ **K6.1: Eine Verknüpfung zu anderen Artefakten ist angemessen möglich.**

Industriestudie. Für die Verknüpfung ist es wichtig, dass das Basismodell geeignet ist, um Abhängigkeiten zwischen den Artefakten in frühen Phasen des Requirements Engineerings angemessen zu repräsentieren, dies wurde in Frage V-GF6 erhoben und mit acht Punkten bewertet.

Die Verbesserung der inneren Struktur von Artefakten wurde in Frage V-GF7 erhoben. Dies wurde mit sechs Punkten bewertet.

Von den Verwendern des Basismodells wurde auch bewertet, wie sich die Verknüpfung von einem Basismodell auf die Qualität der darauf aufbauenden Artefakte auswirkt (Frage V-GF13), dies Verbesserung der Qualität wurde mit sieben Punkten bewertet und daher bestätigt.

Praktikumsstudie. Ein Indiz für die Erfüllung des Kriteriums gibt auch der Kommentar M-K2, welcher die Abdeckung von verschiedenen Arten von Inhalten durch das Basismodell lobt und dadurch, so der Proband weiter, die Erstellung der Use-Cases usw. einfach möglich war.

Darüber hinaus wurde bewertet, ob sich die Qualität der auf dem Basismodell basierenden Artefakte verbessert hat (Frage V-GF13). Dies wurde mit durchschnittlich 7,5 Punkten bestätigt.

- ▶ **K7.1: Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.**

Industriestudie. In Frage V-GF8 haben wir erhoben, ob die Gesamtsystemvorstellung durch das Basismodell klar geworden ist. Dies wurde mit acht Punkten bestätigt.

Praktikumsstudie. Die Entwicklerteams haben bewertet, wie sehr die Gesamtsystemvorstellung durch das Basismodell klar geworden ist. Dies wurde in Frage V-GF8 mit durchschnittlich 7,3 Punkten bewertet.

Einige zusätzliche Hinweise auf das erfolgreiche Schaffen eines stakeholderweiten Verständnisses wurde auch durch die Kommentare T-K2, M-K2 und M-K3 gegeben. Erwähnt wurden, das Schaffen eines guten Rich-Pictures, das gleichzeitige Repräsentieren verschiedener Arten von Inhalten in einem Modell und die gute Kommunikation durch das Modell.

Industriestudie – Anwendbarkeit. Zusätzlich zur Evaluierung haben wir im Fragebogen Fragen vorgesehen, die u.a. auf die Evaluierung der Anwendbarkeit des Modellierungsansatzes abzielen. Von den Verwendern wurde bewertet, ob das Basismodell für den Einsatz in frühen Phasen des Requirements Engineerings geeignet ist (Frage V-GF11). Dies wurde mit sieben Punkten bestätigt. Ein weiterer Aspekt, der in die Anwendbarkeit hineinspielt ist die Einfachheit, Strukturiertheit und Eindeutigkeit des Vorgehens. Dieser Aspekt wurde in den Fragen M-GF1, M-GF2 und M-GF3 mit sieben, acht und sieben Punkten bewertet.

Praktikumsstudie – Anwendbarkeit. Das Vorgehen wurde auf Einfachheit (Frage M-GF1), Strukturiertheit (Frage M-GF2) und Eindeutigkeit (Frage M-GF3) bewertet. Das Ergebnis waren durchschnittlich 6,7, 7,5 und 7,5 Punkte.

Die Anwendbarkeit wurde auch direkt bewertet, dies geschah durch Frage M-GF8 und wurde mit durchschnittlich 7,5 Punkten bewertet.

Ein Aspekt der Anwendbarkeit ist ein minimalistischer Ansatz, der geeignet ist um in frühen Phasen des Requirements Engineerings eingesetzt werden zu können. Dies wurde in Frage V-GF9 mit durchschnittlich sieben Punkten bewertet.

Die Eignung des Basismodells für den Einsatz in frühen Phasen des Requirements Engineerings (Frage V-GF11), wurde mit durchschnittlich 7,8 Punkten bewertet.

Industriestudie – Mehrwert. Wir haben im Rahmen der Verwendung auch gefragt, ob sich der Modellierungsansatz besser eignet, als der zuvor verwendete Ansatz (Frage V-GF12). Dies wurde mit acht Punkten bestätigt.

Die Frage, ob das Basismodell minimalistisch ist (Frage V-GF9) wurde zwar mit fünf Punkten bestätigt, ist aber an der Grenze. Wir haben eine ähnliche Kritik bezüglich des Umfangs auch in der Bewertung in Abschnitt 8.2 festgestellt. Bestätigt wurde dies mit der Aussage, siehe Abschnitt 8.1.5, dass eine Werkzeugunterstützung bei größeren Basismodellen notwendig wird.

Der Ansatz ist nicht fehleranfällig. Dies wurde vom Modellierer des Basismodells in Frage M-GF6 bewertet und mit sechs Punkten bestätigt. Dies ist zwar noch im positiv zutreffenden Bereich, zeigt aber, dass an der Vorgehensweise noch Verbesserungspotential besteht. Dies steht aber auch in einem Spannungsfeld mit der Frage M-GF1, in der die Einfachheit des Modellierungsansatzes mit sieben Punkte bewertet wurde.

Der Modellierer des Modells konnte bewerten, ob die Produktivität in frühen Phasen des Requirements Engineerings durch die Verwendung des Ansatzes aus seiner Sicht gefördert wurde (Frage M-GF5). Dies wurde mit sieben Punkten bewertet. Ein weiterer Hinweis darauf, dass der Ansatz einen Mehrwert bietet sind insbesondere die Kommentare I-K3 und I-K4. Die Kommentare zielten darauf ab, dass (a) die Erstellung des Basismodells zu einer frühen intensiven Diskussion geführt hat und (b), dass das Basismodell als *Leitmodell* sehr hilfreich im Requirements Engineering war.

Praktikumsstudie – Mehrwert. Um den Mehrwert des Modellierungsansatzes für frühe Phasen des Requirements Engineerings zu erheben, haben wir von den Modellierern der Basismodelle bewerten lassen, ob der Ansatz die Produktivität erhöht (Frage M-GF5). Dies wurde mit durchschnittlich 7,3 Punkten bewertet. Die Modellierer sollten zusätzlich bewerten, inwieweit sich der Ansatz in frühen Phasen des Requirements Engineerings eignet (Frage M-GF8). Dies wurde mit durchschnittlich 7,5 Punkten bewertet. Die Frage, ob der Ansatz besser ist, als der zuvor verwendete Ansatz (Frage M-GF9), der typischerweise einer Ad-Hoc-Vorgehensweise entspricht, wurde mit durchschnittlich 7,2 Punkten bewertet.

Eine wichtiger Aspekt in frühen Phasen des Requirements Engineerings ist das Fördern der Kommunikation. Dies wurde von den Modellierern im Erstellungsvorgang (der Modellbildung) mit durchschnittlich 8 Punkten bewertet (Frage M-GF7). Die

gleiche Frage wurde auch den Verwendern gestellt (Frage V-GF10). Und wurde mit durchschnittlich 7,7 Punkten bestätigt.

Ein Kommentar spricht direkt den Mehrwert an (M-K6): „Das Basismodell stellt definitiv einen Mehrwert im Bezug zur bisherigen UML-Modellierung dar.“ Bestätigt wird dies durch die Teambewertungen über die Frage, ob der Modellierungsansatz besser geeignet ist, als der zuvor verwendete (Frage V-GF12). Dies wurde mit durchschnittlich acht Punkten bestätigt.

Ergebnisse im Kontext der vorher verwendeten Vorgehensweise. Die Ergebnisse zeigen, dass der Modellierungsansatz sich gut eignet in frühen Phasen des Requirements Engineerings eingesetzt zu werden. Die positiven Ergebnisse können aber auch dadurch entstanden sein, dass vorher im Requirements Engineering in frühen Phasen keine strukturierte Vorgehensweise eingesetzt wurde (weder in der Industrie noch von den Probanden im Praktikum) und daher jede Form der Repräsentation des Gesamtverständnisses zu diesem Ergebnis geführt hätte. Dass vorher keine strukturierte Vorgehensweise eingesetzt wurde, erklärt auch die Ergebnisse auf die Frage, ob sich der Einsatz des Basismodells in frühen Phasen des Requirements Engineerings besser eignet, als die zuvor verwendete Vorgehensweise (Frage V-GF12).

8.1.8 Validität der Fallstudie

Konstruktvalidität. Die durchgeführten Studien wurden in frühen Phasen des Requirements Engineerings in Softwareentwicklungsprojekten durchgeführt. Der Modellierungsansatz fand daher in der von der Studie geplanten Phase des Entwicklungsprozesses statt.

Den Entwicklungsteams war bekannt, dass es um die Erhebung eines Gesamtverständnisses ging, daher war auch bei den Antworten zu erwarten, dass der Kontext der Fragen richtig interpretiert wurde. Es gab nach allen Fragen die Möglichkeit, bei Unklarheiten Anmerkungen zu machen.

Die entstandenen Artefakte standen nach der Anwendung des Modellierungsansatzes durch die Industriestudie nur teilweise zur Verfügung. In der Praktikumsstudie vollständig. Es wurde überprüft, ob in den Artefakten das Basismodell als Basis verwendet wurde. Dies war der Fall.

Die Fragen zur Bewertung wurden so gestellt, wie sie in Abschnitt 8.1.4 beschrieben wurden. Eine Fehlinterpretation der Fragestellung ist durchaus möglich. Wir gehen davon aus, dass die Wahrscheinlichkeit dass die Fragen falsch verstanden wurden, dennoch gering ist. Da der Ansatz angewendet werden musste und daher alle Schritte, die später durch die Fragen bewertet wurden, den Probanden durchaus bekannt waren, gehen wir davon aus, dass die Fragen von den Probanden auch richtig verstanden wurden.

Interne Validität. Die Ergebnisse der Bewertung werden in Abschnitt 8.1.5 angeführt. Die Fragen in Abschnitt 8.1.4. Die Analyse-Ergebnisse, die aufgrund der erhobenen Artefakte und Bewertungen gezogen wurden, können aufgrund der ange-

fürten Ergebnisse nachvollzogen werden.

In der Praktikumsstudie besteht die Möglichkeit, dass die durch das Basismodell erstellten Artefakte nicht deswegen von einer, für die Probanden, hohen Qualität und untereinander konsistent sind, weil sie auf dem Basismodell basieren, sondern, weil die Probanden die Problemdomäne teilweise kannten.

Zumindest für drei der sechs Projekte kann dies ausgeschlossen werden, da die Problemstellung unabhängig von den Teams erarbeitet wurde und sich dadurch auch die Probanden in die Problemstellung einarbeiten mussten. Da die Ergebnisse der Bewertung dahingehend aber keine Abweichungen aufzeigen, gehen wir davon aus, dass dies den Nutzen des Modellierungsansatzes nicht beeinträchtigt.

Externe Validität.

Industriestudie. Das durchgeführte Entwicklungsprojekt war ein kleines Projekt. Wir wissen nicht, ob der Ansatz auf größere Projekte angemessen angewendet werden kann. Das Projekt hat aber alle Phasen durchlaufen, die im Rahmen eines Softwareentwicklungsprojektes typischerweise durchlaufen werden und bei der Erstellung des Basismodells waren alle Rollen vertreten. Die Größe des Teams, das ein Basismodell erstellt, entspricht dem vorgesehenen Personen-Umfang.

Die Durchführung einer Fallstudie in einem industriellen Projekt, sagt noch nichts über die externe Validität des Modellierungsansatzes aus. Zur Sicherstellung, dass die Verwendung des Modellierungsansatzes den in der Fallstudie gefundenen Mehrwert bietet, wurde die Studie im akademischen Umfeld durchgeführt.

Praktikumsstudie. Auf Grund der Teamgröße, die nicht der durchschnittlichen agilen Projektgröße von ca. 9 Personen [CH01] entspricht, können wir nicht davon sprechen, dass der Ansatz für alle Projektgrößen anwendbar ist. Da die Teamgröße die an der Erhebung beteiligt war aber durch den Ansatz beschränkt wurde und dies auch bei Projekten mit mehr Mitarbeitern der Fall sein würde, wären bei der Erhebung und Erstellung auch nicht mehr Personen beteiligt, als es in der Fallstudie geschehen ist.

Aufgrund der Unerfahrenheit eines Teiles der Probanden, ist es möglich, dass z. B. die Frage nach den Vorhergehenden Ansätzen deswegen zu einem so guten Ergebnis geführt hat, weil es an Erfahrung in diesem Gebiet fehlt. Unabhängig davon hat sich in den Interviews (siehe Kapitel 3) herausgestellt, dass die Systematik in frühen Phasen des Requirements Engineerings vom Unternehmen und den ausführenden Mitarbeitern geprägt ist und es keine explizite Vorgehensweise in frühen Phasen des Requirements Engineerings gibt. Einige der Probanden haben bereits in Softwareprojekten mitgearbeitet und haben ihre Erfahrung in die Bewertung der Verwender (dem Team) miteingebracht.

Zuverlässigkeit.

Industriestudie. Die präsentierten Daten entsprechen den Daten, die wir im Rahmen der Durchführung der Fallstudie erheben konnten. Wir haben versucht, die wichtigsten Artefakte und Ergebnisse zu präsentieren. Leider war uns nur ein Teil der erstellten Artefakte zugänglich. Die Bewertung, auf deren Basis die Analyse durchgeführt wurde, steht vollständig in den Ergebnissen zur Verfügung.

Praktikumsstudie. Alle Ergebnisse der Bewertung sind in Abschnitt 8.1.6.1 repräsentiert. Wir haben nur Teilmengen der Artefakte in Abschnitt 8.1.6.1 eingefügt, die im Rahmen der Projekte entstanden sind. Diese dienen der Anschauung. Wir haben versucht, die Ergebnisse so zu repräsentieren, dass auch dritte Personen unsere Schlussfolgerungen nachvollziehen können.

8.2 Bewertung des Modells unabhängig von einem Softwareprojekt

Im Rahmen der Evaluierung des Basismodells wurde dieses, unabhängig von Softwareentwicklungsprojekten und dessen Verwendung in diesen, bewertet. In diesem Abschnitt wird die Bewertung dargestellt.

8.2.1 Kontext

Wir nehmen die Einteilung des Kontextes nach den Vorgaben von Wohlin et al. [WRH⁺12] vor.

Selbständig. Das Modell wurde selbständig und ohne initiale Erklärung vom Experten analysiert.

Professioneller Kontext. Das Modell soll in einem professionellen Kontext bewertet werden.

Reales Problem. Das übermittelte Basismodell stammt aus einem realen Softwareprojekt.

Allgemeines Beispiel. Das übermittelte Beispiel ist für ein spezielles Problem erstellt worden, die Darstellung der Entitäten und Beziehungen entspricht aber der typischer Basismodelle.

8.2.2 Forschungsziel

Wir verwenden zur Formulierung des Forschungszieles die Schablone von Wohlin et al. [WRH⁺12].

- ▶ **Analysiert wird:** Das Basismodell

- ▶ **Für den Zweck:** Der Eignung
- ▶ **Im Bezug auf:** Lesbarkeit, Verständlichkeit, Kommunizierbarkeit
- ▶ **Aus Sicht von:** Eines Prozess-Analysten
- ▶ **Im Kontext:** Eines prozessorientierten Unternehmens

8.2.3 Aufbau und Erhebung

Folgende geschlossene Fragen wurde im Rahmen der Erhebung gestellt:

GF1 Die im Modell repräsentierten Entitäten sind eindeutig: Eindeutig sind die Entitäten dann, wenn ihre Bedeutung durch die Stereotypisierung und den Beziehungen zu anderen Entitäten für Sie eindeutig sind.

GF2 Die im Modell repräsentierten Entitäten sind verständlich: Ein Modell ist verständlich, wenn es einfach lesbar ist, nur eine geringe qualitative Komplexität aufweist (=inhaltliche Komplexität) und die Bedeutung der Entitäten spätestens durch die Entitäten und deren Beziehungen klar ist.

GF3 Das Modell ist angemessen: Angemessen ist ein Modell, wenn es für seinen Zweck geeignet ist. Der Zweck des Modells: Repräsentation eines Gesamtverständnisses (bezeichnet das Verständnis über Geschäft und unterstützende Software), Repräsentation eines stakeholderweiten Verständnisses, und der Nutzen als Basis für weitere Artefakte (z. B. Geschäftsprozesse, die diese im Modell detaillieren).

Hier sei angemerkt, dass nur jene Facetten des Zweckes in der Evaluierung vorkommen, die für den Kontext der Fallstudie relevant waren.

GF4 Modell ist zur Kommunikation im Geschäftsbereich geeignet: Glauben Sie, dass das Modell für die Kommunikation im Geschäftsbereich, z. B. in ihrem Unternehmen geeignet ist, um die Unterstützung der Software in Hinblick auf die Geschäftsprozesse zu repräsentieren?

GF5 Modell ist zur Kommunikation über ihren Geschäftsbereich hinaus geeignet: Glauben Sie, dass das Modell über ihre Geschäftstätigkeit hinaus zur Kommunikation geeignet ist? Z. B. Sie wollen Prozesse mit Softwareanteilen unterstützen und geben dazu das Modell an Softwareentwickler weiter, damit diese Wissen, was sie unterstützt haben möchten.

GF6 Modell ist zur Klärung von inhaltlichen Missverständnissen geeignet: Glauben Sie, dass das Modell zur Klärung von Missverständnissen in ihrem Unternehmen in Hinblick auf den Einsatz von Softwareanteilen in Geschäftsprozesse geeignet ist?

GF7 Das Modell ist lesbar und verständlich ohne zusätzliche Beschreibung: Glauben Sie, dass das Modell ohne zusätzliche Beschreibung lesbar und verständlich ist?

Die geschlossenen Fragen boten Antwortmöglichkeiten in einer achtstufigen Skala von *trifft völlig zu* bis *trifft überhaupt nicht zu*. Es wurde eine achtstufige Skala verwendet, um eine neutrale Aussage zu vermeiden.

Die Fragebögen boten nach jeder Frage die Möglichkeit, freie Anmerkungen zu machen.

Folgende offenen Fragen wurden im Anschluss an die Erhebung in einem Interview gestellt:

OF1 Ist das Modell lesbar und verständlich ohne zusätzliche Beschreibung?

Diese Frage wurde offen und geschlossen gestellt, um qualitativ bessere Informationen zu erhalten.

OF2 Kann das Modell zur unternehmensweiten und -übergreifenden Kommunikation verwendet werden?

Diese Frage wurde offen und geschlossen gestellt, um qualitativ bessere Informationen zu erhalten.

OF3 Welche Nachteile sehen Sie?

Erhebung. Zuerst wurde das Modell dem Probanden übermittelt. Der Proband konnte sich mit dem Modell beschäftigen. Es wurden, bis auf das Modell, keine Informationen übermittelt. Übermittelt wurde das Modell, das im Rahmen der Fallstudie Industrie erstellt wurde (siehe Abbildung 8.4).

Der Proband wurde danach gebeten, das Modell anhand des vorher angeführten Fragebogens zu bewerten. Anschließend wurde ein Tiefeninterview durchgeführt, das auf spezielle Aspekte noch weiter eingeht. Wir wollten damit über die Bewertung hinaus, nach einer vorgegebenen Skala, noch qualitative Informationen erheben.

Subjekt- und Fallauswahl. Um das Modell im Kontext einer unternehmerischen Tätigkeit bewerten zu können, mussten wir einen Probanden finden, der als Experte gilt und bereit war, das Basismodell zu bewerten. Wir haben als Experten all jene Personen gesehen, die über einige Jahre Erfahrung im Bereich Geschäftsprozessmanagement verfügen. Die von uns befragte Person ist Experte mit langjähriger Erfahrung und akademischen Abschlüssen im Umfeld *Operations Management* und *Supply Chain Management*.

Der übermittelte Fall stammt aus der Praxis. Dazu wurde das im Rahmen der Fallstudie Industrie erstellte und uns zur Verfügung gestellte Basismodell aus Iteration 4 verwendet (siehe Abbildung 8.4).

8.2.4 Ergebnisse

Abbildung 8.15 zeigt die Bewertung der geschlossenen Fragen durch den Probanden.

Kommentare zur Lesbarkeit (OF1):

Der Proband hat dazu versucht, was allgemein im Basismodell herauszulesen ist, zu beschreiben:

K1 „Das Modell beschreibt u.a. Teile eines Softwaresystems. Das Softwaresystem besteht aus Teilen.“

	trifft völlig zu			trifft überhaupt nicht zu			
GF1		X					Inhalt ist eindeutig
GF2			X				Das Modell ist verständlich
GF3	X						Das Modell ist angemessen
GF4		X					Modell ist zur internen Kommunikation geeignet
GF5			X				Modell ist zur übergreifenden Kommunikation geeignet
GF6				X			Modell ist zur Klärung von inhaltl. Missverständnissen geeignet
GF7		X					Model ist ohne Beschreibungen lesbar und verständlich

Abbildung 8.15: Bewertung durch domänenferne Person

- K2 „Das Softwaresystem ist nach Unterstützungsfunktionen aufgeteilt.“
- K3 „Das Modell stellt Geschäftsprozesse dar. Auch Geschäftsprozesse werden in Teilprozesse aufgeteilt.“
- K4 „Geschäftsprozesse verursachen andere Geschäftsprozesse und haben Einfluss auf Geschäftsobjekte.“
- K5 „Welchen Einfluss ein Geschäftsprozess auf ein Geschäftsobjekt hat, ist durch die Beziehungen erkennbar. Dies geschieht verständlich und eindeutig.“
- K6 „Die Unterstützungsfunktionen sind mit Geschäftsprozessen verbunden. Man erkennt, welche Unterstützungsfunktionen welche Geschäftsprozesse unterstützen.“

Kommentare zur unternehmensinternen und -übergreifenden Kommunikation (OF2):

- K7 „Modell ist gut zur Kommunikation geeignet. Besser als textuelle Beschreibungen.“
- K8 „Das Modell bekommt einen besonderen Wert, da es die Rolle der Software, im Gegensatz zu anderen Geschäftsprozessmodellierungstechniken nicht stiefmütterlich behandelt, sondern als den wichtigen Teil eines Unternehmens, der Software zweifelsohne ist, darstellt.“
- K9 „Das Modell hilft den dargestellten Sachverhalt wesentlich besser zu kommunizieren, wie es bei reinen Darstellungen der Prozesse der Fall ist.“
- K10 „Ich habe mich schon öfter gefragt, wie man die Menge von Prozessen in einem Unternehmen und die involvierten Daten kommunizieren kann. Dies ist eine gute Möglichkeit und fokussiert auf wichtige Aspekte.“

Kommentare zu Nachteilen (OF3):

- K11 „Ich kann mir vorstellen, dass irgendwann zu viele Entitäten vorhanden sind.“
- K12 „Geschäftsprozesse müssen anscheinend keine Ordnung haben.“
- K13 „Nicht genügend Details für unternehmensübergreifende Kommunikation.“

Aber das ist vom Ziel abhängig.“

8.2.5 Analyse

Wir analysieren die erhobenen Antworten und erstellten Basismodelle in Diskussion anhand der Kriterien, die wir in Kapitel 5 aufgestellt haben. Da die Experten-Befragung nicht alle Kriterien adressiert, sind nur die angeführt, die adressiert werden.

Eine zusammenfassende Diskussion der Ergebnisse findet in Abschnitt 8.4 statt.

Die Bewertung, wie im Aufbau in Abschnitt 8.2.3 erwähnt, erfolgte in einer achtstufigen Skala, von *Trifft völlig zu* bis *Trifft überhaupt nicht zu*. Wir sprechen im Folgenden von acht Punkten, wenn die Bewertung *Trifft völlig zu* gewählt wurde und von einem Punkt, wenn *Trifft überhaupt nicht zu* gewählt wurde. Die achtstufige Skala erlaubte es uns neutrale Aussagen zu vermeiden. Wir sprechen daher ab 5 Punkten und mehr, von einer eher zutreffenden Aussage und bei 4 Punkten und weniger, von einer eher nicht zutreffenden.

► **K1.1: Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.**

Die Interpretation des Modells wird in Abschnitt 8.3 behandelt. Wir beschäftigen uns im Kriterium mit der Frage, ob das Basismodell helfen kann, inhaltliche Missverständnisse zu klären.

Wir haben in Frage GF1 erhoben, wie eindeutig, für den Leser eines Basismodells, die Inhalte im Sinne einer subjektiven Interpretation sind. Dies wurde mit sieben Punkten bewertet.

Ähnlich wichtig ist die Frage nach der Verständlichkeit eines Basismodells. Das Basismodell soll daher eine geringe qualitative Komplexität aufweisen. Dies wurde in Frage GF2 mit sechs Punkten bewertet.

Die Interpretation eines Basismodells wird in Abschnitt 8.3 noch genauer untersucht.

Die Frage, ob das Basismodell zur Klärung von inhaltlichen Missverständnissen geeignet ist (GF6) wurde mit fünf Punkten bewertet.

Besonders hervorgehoben wurde in Kommentar K8, dass das Basismodell einen besonderen Wert hat, da die Rolle der Software nicht stiefmütterlich behandelt wird, sondern als wichtiger Teil eines Unternehmens, im Modell auch entsprechend repräsentiert ist.

► **K7.1: Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.**

Besonders wichtig für die Arbeit ist die Bewertung der Angemessenheit des Modells für den Zweck der Repräsentation des Gesamtverständnisses. Dies wurde durch Frage GF3 mit acht Punkten bewertet.

In Frage GF4 wurde erhoben, ob das Basismodell zur internen Kommunikation geeignet ist. Dies wurde mit sieben Punkten bewertet. Die Eignung des Basismodells zur übergreifenden Kommunikation in Frage GF5 wurde mit sechs Punkten bewertet.

Die Lesbarkeit des Basismodells für die Weitergabe ist eine wichtige Eigenschaft. Dies wurde in Frage GF7 mit sieben Punkten bewertet.

In Kommentar K13 wurde angemerkt, dass für eine unternehmensübergreifender Kommunikation zu wenig Details im Modell sind, dies aber vom Ziel abhängt. Andererseits wurde in Kommentar K9 erwähnt, dass das Basismodell hilft, Sachverhalte wesentlich besser zu kommunizieren, als eine reine Darstellung von Geschäftsprozessen. Auch Kommentar K7, dass das Basismodell sehr gut zur Kommunikation geeignet ist, unterstreicht unsere Aussage.

8.2.6 Validität der Fallstudie

Konstruktvalidität. Die Konstruktion des Fragebogens erfolgte anhand der Qualitätskriterien. Die Bewertung von domänenfernen Personen schien uns hier besonders hilfreich, da diese keinen Bezug zu softwarespezifischen Artefakten hatten.

Die Fragen, wie in Abschnitt 8.2.3 ersichtlich, wurden detailliert, damit der Proband die Frage nicht falsch verstehen konnte. Nach jeder Frage gab es die Möglichkeit, auf Unklarheiten hinzuweisen. Zusätzlich haben wir im Anschluss ein Tiefeninterview durchgeführt, auch dort gab es noch die Möglichkeit, offene Fragen zu klären.

Das übermittelte Modell stammt aus einer Domäne, die dem Proband nicht bekannt war.

Interne Validität. Die Ergebnisse der durchgeführten Bewertung werden in Abschnitt 8.2.4 vollständig offen gelegt.

Externe Validität. Die durchgeführte Studie bewertet das Basismodell in Hinblick verschiedener Eigenschaften innerhalb eines Unternehmens durch einen Experten. Der Experte stellte dar, dass viele der bewerteten Eigenschaften, abhängig vom Ziel zu bewerten sind. Hier muss in zukünftigen Studien die Zielsetzung noch klarer kommuniziert werden.

Zuverlässigkeit. Der befragte Proband erhielt das Modell nur mit der Bitte, dieses zu analysieren und anschließend auf verschiedene, im Abschnitt, eingeführten Kriterien zu bewerten. Der Proband wusste nicht, ob wir uns negative oder positive Antworten erwarten. Der Proband konnte auch nicht wissen, ob das Modell auf Basis des Modellierungsansatzes, vorgestellt in dieser Arbeit, erstellt wurde und seine Antworten zur Evaluierung dienten.

8.3 Laborexperiment

Im Rahmen der Evaluierung der Erfüllung von Kriterien der Problemanalyse (siehe Kapitel 5) haben wir mit acht ausgewählten Probanden ein Laborexperiment durchgeführt. Mit dem Experiment sollen die Daten, die nicht im Rahmen der Fallstudien erhebbar waren, erfasst werden.

8.3.1 Kontext

Wir beschreiben den Kontext des Laborexperiments nach den Vorgaben von Wohlin et al. [WRH⁺12].

Selbständig. Das durchgeführte Experiment wurde von den Probanden eigenständig durchgeführt. Es gab eine kurze Einführung zum Modus (siehe Design und Beschreibung des Ablaufes). Diesem Modus musste während der Bearbeitung gefolgt werden.

Professionelle Teilnehmer. Alle Teilnehmer, siehe dazu Subjekt-Auswahl, haben Erfahrung in Softwareentwicklungsprojekten.

Aufgaben-Größe. Die Aufgaben, die im Rahmen des Experimentes zu erledigen waren, sind kleine Aufgaben. Sie entsprechen daher typischen *Toy-Examples* [WRH⁺12].

Speziell. Die im Rahmen des Experimentes zu analysierenden Fragestellungen, siehe Abschnitt 8.3.3, sind spezielle Fragestellungen, die auf die zu evaluierenden Kriterien zugeschnitten wurden und daher außerhalb des Experimentes keine Aussagekraft haben.

8.3.2 Forschungsziel

Wir verwenden zur Formulierung des Forschungszieles die Schablone von Wohlin et al. [WRH⁺12].

- ▶ **Analysiert werden:** Verschiedene Aspekte bei der Verwendung des Basismodells
- ▶ **Für den Zweck:** Evaluation
- ▶ **Im Bezug auf:** Interpretation eines Basismodells und Modellbildung
- ▶ **Aus Sicht von:** Softwareentwickler und Anforderungsingenieur
- ▶ **Im Kontext:** Experiment

8.3.3 Forschungsfragen

Im Rahmen des Experiments soll ein Beitrag zur Beantwortung von Forschungsfrage 4 geleistet werden.

RQ4: Inwiefern hilft der erarbeitete Ansatz bei der Lösung der Probleme bezüglich RQ3 bei der Erhebung von Problem- und Lösungsdomäne in frühen Phasen des Requirements Engineerings?

Um die Forschungsfrage beantworten zu können, müssen die einzelnen Kriterien, die im Rahmen der Beantwortung von Forschungsfrage 3 gefunden wurden, adressiert werden.

RQ3: Welche Probleme treten bezüglich RQ1 und RQ2 bei den Überbrückungsansätzen in der Praxis auf?

Die Forschungsfrage 3 wird in Kapitel 5 beantwortet. Nicht alle in Kapitel 5 erarbeiteten Kriterien, konnten im Rahmen der Studie evaluiert werden.

Im Rahmen des Experiments wird ein Beitrag zur Evaluierung folgender Kriterien³ geleistet:

- ▶ **K1.1:** Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.

Im Laborexperiment wollen wir untersuchen, inwieweit ein Basismodell von verschiedenen Probanden *gleich* interpretiert wird. Wir sprechen von gleicher Interpretation, wenn die von den Probanden explizierte Interpretation von einer dritten Person, die die Interpretationen liest, als gleich identifizierbar ist.

Im Laborexperiment stellen wir auch Fragen, die die Eignung zur Klärung von Missverständnissen bewerten. Wir wollen hier einen weiteren Beitrag zur Evaluierung der Erfüllung dieses Kriteriums leisten.

- ▶ **K4.1:** Die Artefakte, die gleiche Ausschnitte eines Problems bzw. einer Lösung mit Hilfe der gleichen Modellierungstechnik beschreiben, werden von unterschiedlichen Personen ähnlich modelliert.

Im Rahmen des Laborexperiments wollen wir untersuchen, inwieweit ein geschilderter Sachverhalt von unterschiedlichen Personen, durch die Anwendung der Vorgehensweise, *gleiche* Modelle produziert.

Im Laborexperiment wollen wir mit weiteren Fragen, siehe dazu Abschnitt 8.3.4, auch die Eignung der Vorgehensweise bzw. der repräsentierten Inhalte in der Modellbildung evaluieren.

- ▶ **K7.1:** Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.

Im Rahmen des Laborexperimentes wollen wir einen Beitrag zur Evaluierung dieses Kriteriums leisten. Wir bewerten verschiedene Aspekte zur Kommunikation und Verständlichkeit von modellierten Inhalten in einem Basismodell.

8.3.4 Aufbau des Experiments und Erhebung

Fall und Subjekt-Selektion Zur Beantwortung der Forschungsfragen war es notwendig, Personen auszuwählen, die (a) mit Softwareentwicklungsprojekten eingehend vertraut waren und (b) bereits Erfahrung mit Modellierungsansätzen hatten.

Mit acht ausgewählten Probanden wurde im Rahmen dieser Arbeit ein Experiment durchgeführt. Die Probanden mussten zwei Voraussetzungen erfüllen: (a) sie muss-

³ Die Kriterien stammen aus Kapitel 5.

ten bereits in einem Softwareprojekt gearbeitet haben bzw. arbeiten, in Rahmen dessen Informationssysteme entwickelt wurden bzw. werden und (b) sie mussten mit den UML-Diagramm-Typen vertraut sein.

Aufbau des Experiments. Das Experiment besteht aus einer Menge von Fragen und Aufgaben. Wir unterscheiden Ja/Nein-Fragen (JNF), Skala-Fragen (SF)⁴, offene Fragen (OF) und Aufgaben (AG). Nach allen Ja/Nein- und Skala-Fragen (bis auf JNF1) gab es die Möglichkeit Anmerkungen zu machen. Diese werden an gegebener Stelle in den Ergebnissen angeführt.

Aufbau des Experiments:

JNF1 Ist Ihnen der in dieser Arbeit vorgestellte Modellierungsansatz bekannt?

AG2 Beschreiben Sie in eigenen Worten den folgenden Sachverhalt so detailliert wie möglich.

Im Rahmen der Aufgabe wurde versucht herauszufinden, wie ein Basismodell von den Probanden interpretiert wird, wenn sie ein solches zum ersten Mal sehen. Dazu wurde ein kleines Basismodell in die Aufgabe integriert.

OF2b Welche Bedeutung haben die Stereotypen im Modell?

JNF3 Hatten Sie eine eigene Vorstellung von dem in AG2 vorgestellten Sachverhalt?

SF4 Finden Sie, dass Ihre Vorstellung über Aufbau und Ablauf des geschilderten Sachverhaltes angebracht abgebildet ist?

SF5 Finden Sie, dass dieses Modell der Kommunikation zwischen verschiedenen, in Projekten involvierten Stakeholdern besser dient, als es der gleiche Sachverhalt in einem UML-Klassendiagramm getan hätte?

AG6 Bitte beschreiben Sie den folgenden Sachverhalt. Beschreiben Sie, was sie aus dem Modell mit Sicherheit herauslesen können (Aufgabe).

Im Rahmen der Aufgabe wird ein Basismodell präsentiert, das einen Ausschnitt einer Domäne beschreibt, die den Probanden unbekannt sein sollte. Mit diesem Experiment soll versucht werden herauszufinden, ob die Interpretation von einer möglicherweise bekannten Domäne (aus AG2) abweicht.

OF6b Was könnten die Beziehungen zwischen den grauen und schwarzen Modell-elementen bedeuten?

Im Rahmen dieser Frage wird erhoben, ob die Probanden die Trennung zwischen Problem- und Lösungsdomäne, ohne explizite Erklärung, verstehen.

JNF7 Hatten Sie eine eigene Vorstellung vom geschilderten Sachverhalt in AG6?

SF8 Halten Sie die Trennung zwischen Softwareanteilen und Geschäftsbereichswissen in frühen Phasen des Requirements Engineerings für sinnvoll?

SF9 Hat die in SF8 angesprochene Trennung Ihre Vorstellung über den Gesamtzusammenhang präzisiert?

SF10 Glauben Sie, dass das Modell besser geeignet ist, die Trennung zu ermöglichen,

⁴ Wir bezeichnen als Skala-Fragen all die Fragen, die in einer achtstufigen Skala von *trifft völlig zu* bis *trifft überhaupt nicht zu* beantwortet werden können.

als es z. B. UML-Klassendiagramme erlauben?

AG11 Eigenes Basismodell.

Im Zuge der Erstellung haben alle Teilnehmer eine kurze Beschreibung des Vorgehens sowie der Konzepte und Beziehungen erhalten, die mit einem Basismodell erfassbar sind. Alle Probanden mussten sich daher im Rahmen der Aufgabe in den Modellierungsansatz einarbeiten.

In der Aufgabe wurde eine Problemdomäne beschrieben, die von einem Softwaresystem unterstützt werden soll. Die Probanden mussten die relevanten Terme extrahieren und mit Hilfe der Vorgehensweise und den zur Verfügung gestellten Konzepten und Beziehungen modellieren.

SF11b Konnten Sie alle relevanten Sachverhalte zu ihrer Zufriedenheit abbilden?

SF11c Hatten Sie eine eigene Vorstellung von der geschilderten Domäne?

Hier wurde eine Skala-Frage gewählt, da es durchaus möglich war, dass teilweise eine Vorstellung von der geschilderten Domäne vorhanden ist, aber nicht ausgeprägt genug, um diese eigenständig zu beschreiben.

SF12 Konnten Sie den geschilderten Sachverhalt zielorientierter abbilden, wie es mit einem UML-Klassendiagramm möglich wäre?

Zielorientiert möglich ist eine Abbildung dann, wenn es geeignete primitive Modellelemente gibt, die die Abbildung eines Sachverhaltes direkt und angemessen ermöglichen.

SF13 Haben die vordefinierten Konzepte und Beziehungen Ihnen geholfen, Ihr Verständnis über die Begriffe der Domäne zu präzisieren?

SF14 Glauben Sie, dass das von Ihnen in AG11 erstellte Modell sich eignet, um die Gesamtvorstellung im Projekt und darüber hinaus zu kommunizieren?

Fall- und Subjektauswahl. Um die offenen Fragen beantworten zu können, mussten wir Softwareentwickler finden, die den Modellierungsansatz, vorgestellt in dieser Arbeit nicht kannten und Erfahrung in Softwareentwicklungsprojekten haben. Dies war notwendig, um Aussagen über die Aussagekraft des Experiments treffen zu können.

Alle gewählten Softwareentwickler mussten auch Erfahrung mit den UML-Modell-Typen haben. Dieser Vergleich ist notwendig, da in frühen Phasen des Requirements Engineerings häufig UML-Modelle erstellt werden, die nicht integriert werden und sich daher ein Vergleich mit diesen Modellierungstechniken anbot.

Ablauf des Experiments. Jeder Proband bekam zwei Umschläge, im ersten Umschlag war der erste Teil des Experiments. Der erste Teil bestand aus den Fragen bzw. Aufgaben von 1 bis 10. Nach Vollendung der Fragen bzw. Aufgaben 1 bis 10 durften die Probanden den zweiten Umschlag öffnen, der enthielt die Fragen bzw. Aufgaben 11 bis 14. Im zweiten Umschlag war auch eine kurze Einführung des Modellierungsansatzes enthalten, den die Probanden zur Beantwortung der Aufgabe 11 benötigten. Die Aufteilung des Experimentes in zwei Umschläge war notwendig, da das vorzei-

tige Lesen der kurzen Einführung in den Modellierungsansatz die Beantwortung der Aufgaben 1 bis 10 beeinflusst hätten.

8.3.5 Ergebnisse

Dieser Abschnitt enthält die während des Experimentes erhobenen Daten und Modelle.

Ergebnisse der geschlossenen Fragen. Abbildung 8.16 zeigt die Ergebnisse der Ja- und Nein-Antworten.

	Ja	Nein	
JNF1	0	8	Ansatz ist bekannt
JNF3	6	2	Eigene Vorstellung vom geschilderten Sachverhalt vorhanden
JNF7	2	6	Eigene Vorstellung vom geschilderten Sachverhalt vorhanden

Abbildung 8.16: Ergebnis der Ja- und Nein-Antworten

Abbildung 8.17 zeigt das Ergebnis der Antworten mit Skala.

	trifft völlig zu				trifft überhaupt nicht zu				
SF4	2	1	4	1	0	0	0	0	Angebrachte Repräsentation d. Vorstellung
SF5	2	3	2	1	0	0	0	0	Zur Kommunikation besser geeignet
SF8	4	2	0	1	0	0	1	0	Integrative Darstellung ist sinnvoll
SF9	2	3	2	1	0	0	0	0	Vorstellung über Zusammenhang präzisiert
SF10	2	2	2	2	0	0	0	0	Trennung führt zu mehr Präzision
SF11b	3	3	2	0	0	0	0	0	Abbildung rel. Sachverhalte möglich
SF11c	0	1	0	0	0	1	0	6	Domäne war bekannt
SF12	2	4	2	0	0	0	0	0	Präzise Abbildung war möglich
SF13	6	1	1	0	0	0	0	0	Vord. Konzepte und Beziehungen helfen
SF14	3	4	0	0	0	0	0	1	Kommunikation im Projekt und darüber hinaus

Abbildung 8.17: Ergebnis der Antworten mit Skala

AG2: Interpretation eines kleinen Beispiels. Abbildung 8.18 zeigt das im Rahmen von AG2 verwendete minimale Basismodell ohne Softwareanteile.

Im Rahmen der Beantwortung von OF2b wurden einige Antworten gegeben. Nachfolgend werden diese Antworten gekürzt angeführt. Beantwortet werden sollte die Frage, was die Stereotypen im Modell aussagen.

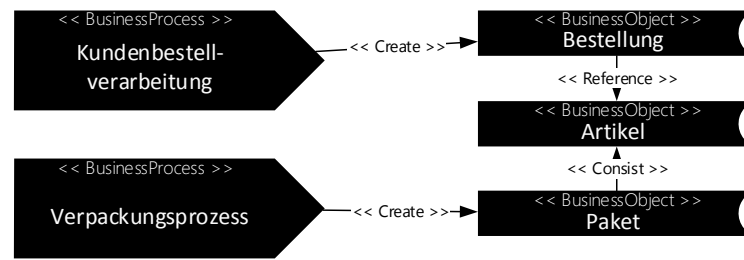


Abbildung 8.18: AG2: Basismodell einer eher bekannten Domäne

- ▶ „[...] sie verraten uns, dass es sich um den Geschäftsbereich handelt [...]“
- ▶ „[...] ein Geschäftsprozess kann andere anstoßen und Geschäftsobjekte involvieren [...] stellt etwas Reales dar [...]“
- ▶ „[...] zeigen an, wie Prozesse ablaufen und wie Objekte involviert sind [...]“
- ▶ „[...] Vorgänge und Geschäftsobjekte voneinander trennen [...]“

Angemerkt sei, dass in den Zitaten *BusinessProcess* in (Geschäfts-)Prozess und *BusinessObject* in Geschäftsobjekt übersetzt wurden.

In JNF3 wurde erhoben (siehe Abbildung 8.16), ob die geschilderte Domäne bekannt war.

AG6: Interpretation einer eher unbekanntem Domäne. Abbildung 8.19 zeigt das Basismodell zur AG6, einer eher unbekanntem Domäne.

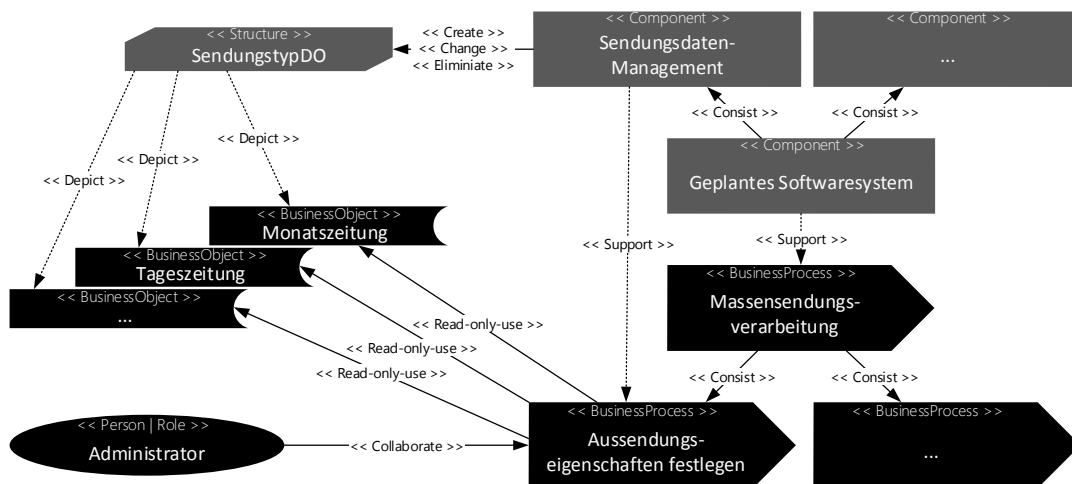


Abbildung 8.19: AG6: Basismodell einer allgemein nicht bekannten Domäne

In JNF7 wurde erhoben (siehe Abbildung 8.16), dass die Domäne für die Probanden unbekannt war.

Nachstehend führen wir Ausschnitte aus der Interpretation des Modells in Abbildung 8.19 an:

- ▶ „Im Lösungsbereich dient die Komponente *Sendungsdaten-Management* als *Verwalter* für alle Objekte/Strukturen vom Typ *SendungstypDO*. Die Struktur ist

nur eine Abbildung vom Geschäftsobjekt *Sendungstyp*.“

- ▶ „Das Softwaresystem beinhaltet eine vom Administrator verwaltete Struktur von *SendungstypenDO*. Jedes Modell stellt generische Erwartungen auf, die von einem konkreten Geschäftsobjekt erfüllt werden.“
- ▶ „Das *Sendungsdaten-Management* verwaltet die verschiedenen *SendungstypenDOs*, die Geschäftsobjekte abbilden.“
- ▶ „*Sendungsdaten-Management* unterstützt den Geschäftsprozess *Aussendungseigenschaften festlegen* und erstellt, ändert und löscht *SendungstypenDOs*. *SendungstypenDOs* bilden Geschäftsobjekte ab.“
- ▶ „Das *Sendungsdaten-Management* ist für die Erstellung, Änderung und Löschung von *SendungstypenDO* zuständig.“
- ▶ „Der Prozess *Aussendungseigenschaften festlegen* nutzt ein oder mehrere *Sendungstypen* im Prozess. Der Prozess ist nicht dazu fähig neue *Sendungstypen* zu erzeugen.“
- ▶ „Der Prozess *Aussendungseigenschaften festlegen* wählt einen (oder mehrere) *Sendungstypen* im Prozess.“
- ▶ „Der Prozess *Aussendungseigenschaften festlegen* verwendet die *Sendungstypen* im Prozess.“
- ▶ „*Aussendungseigenschaften festlegen* benutzt das Geschäftsobjekt *Sendungstyp*. Es im Prozess verwendet.“
- ▶ „*Aussendungseigenschaften festlegen* benutzt den *Sendungstyp* um Eigenschaften für die *Sendung* festzulegen.“

Nachstehend sind die Antworten angeführt, die auf OF6b gegeben wurden. Es ging um die Bedeutung der Übergänge zwischen den schwarzen und grauen Modellelementen.

- ▶ „[...] Die Beziehungen deuten an, dass Geschäftsprozesse durch Softwarekomponenten unterstützt werden und, dass die Struktur *SendungstypDO* einem realen Geschäftsobjekt softwareseitig entspricht. [...]“
- ▶ „[...] *Depict*: Eine Struktur bildet echte Geschäftsobjekte ab. *Support*: Ein Softwaresystem unterstützt einen Geschäftsprozess.“
- ▶ „[...] Die Beziehungen repräsentieren den Übergang von Software- und Geschäftsbereich. Das Wort *Support* beschreibt, wie der Geschäftsbereich vom Softwarebereich unterstützt wird. Durch *Depict* wird angegeben, dass *Structure* im Lösungsbereich Geschäftsobjekte im Geschäftsbereich repräsentieren.“
- ▶ „Grau: Domäne der IT-Welt, schwarz: Domäne der konkreten Einsatzumgebung. Beziehungen: Verknüpfung beider Welten.“
- ▶ „Die Verbindung zwischen der Umsetzung/Programmierung und der Geschäfts- bzw. Prozesssicht auf den Sachverhalt. Sehr positiv ist, dass es ein Modell gibt, das von der Fachabteilung und von Entwicklern verstanden wird.“
- ▶ „[...] Komponenten sollen Prozesse unterstützen.“

Angemerkt sei, dass in den Zitaten *BusinessProcess* in (Geschäfts-)Prozess, *Business-*

Object in Geschäftsobjekt und Structure in Struktur übersetzt wurde.

AG11: Eigenes Basismodell aus Beschreibung. Abbildung 8.20 zeigt vier der acht im Rahmen des Experiments erstellten Basismodelle.

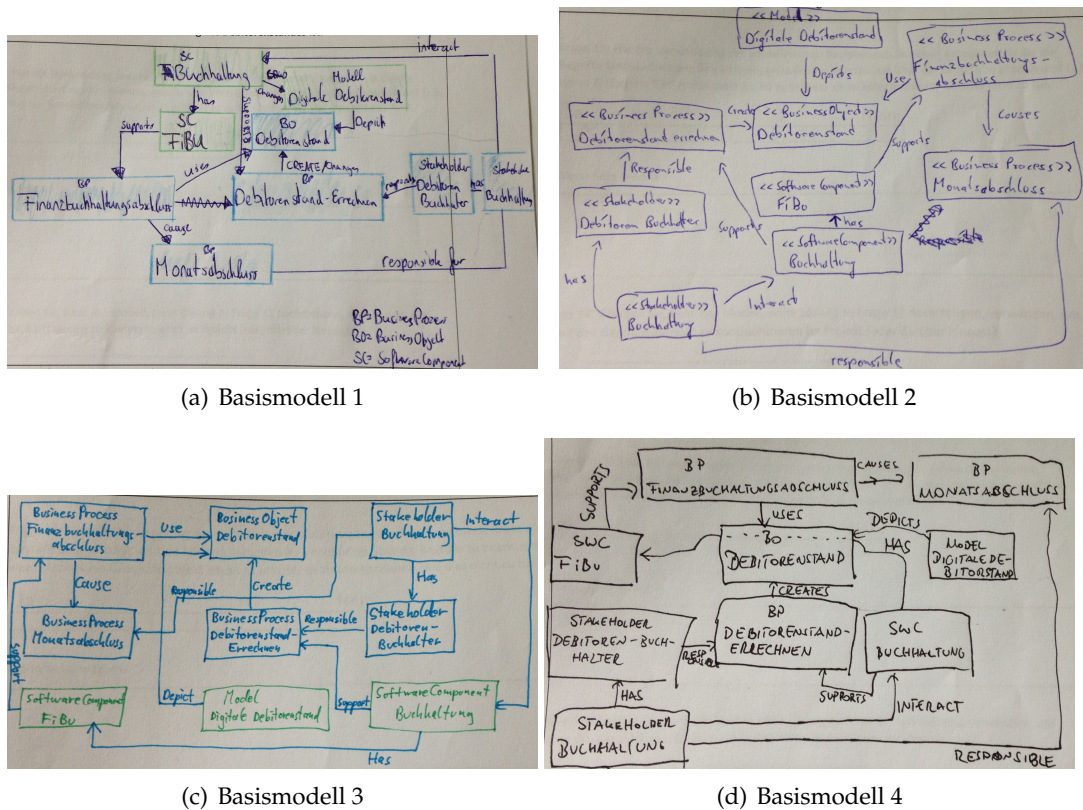


Abbildung 8.20: Ausschnitt aus den während des Experimentes erstellten Basismodellen

Ergebnisse Vergleich: In sieben der acht Modelle wurden alle relevanten Entitäten identifiziert. In diesen sieben Modellen wurden auch alle Entitäten dem richtigen Konzept zugeordnet.

In einem Modell wurde bei einigen Entitäten kein Konzept zur Stereotypisierung angegeben.

Kommentare von Probanden nach dem Experiment.

K1 „Ich hatte das Problem in der Praxis häufig, dass BWLer mit der Modellierung von Prozessen begannen und die Softwareentwickler mit der Erstellung von UML-Klassendiagrammen, die Idee des Ansatzes werde ich, etwas abgeändert, auch in der Praxis in Zukunft einsetzen.“

K2 „Damit dein Ansatz verwendet wird, musste er einfach und auf das Wesentliche beschränkt sein, das hast du geschafft und ich werde ihn verwenden.“

K3 „Sehr positiv ist, dass es ein Modell gibt, dass von der Fachabteilung und von Entwicklern verstanden wird.“

8.3.6 Analyse

Eine zusammenfassende Diskussion der Ergebnisse findet in Abschnitt 8.4 statt.

Um sicherzugehen, dass das Experiment nicht vom Wissen über den Modellierungsansatz, der in dieser Arbeit vorgestellt wird, beeinflusst wird, wurde als erste Frage nach dem Bekanntsein des Ansatzes gefragt. Alle Probanden kannten den Ansatz vor dem Experiment nicht (siehe Abbildung 8.16, JNF1).

Die Bewertung, wie im Aufbau in Abschnitt 8.3.4 erwähnt, erfolgte bei den SF-Fragen in einer achtstufigen Skala, von *Trifft völlig zu* bis *Trifft überhaupt nicht zu*. Wir sprechen im Folgenden von acht Punkten, wenn die Bewertung *Trifft völlig zu* gewählt wurde und von einem Punkt, wenn *Trifft überhaupt nicht zu* gewählt wurde. Die achtstufige Skala erlaubte es uns neutrale Aussagen zu vermeiden. Wir sprechen daher ab 5 Punkte und mehr, von einer eher zutreffenden Aussage und bei 4 Punkten und weniger, von einer eher nicht zutreffenden.

Wir analysierten die erhobenen Antworten und erstellten Basismodelle in Diskussion mit den Kriterien.

► **K1.1: Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.**

In den Ergebnissen werden Ausschnitte aus den Interpretationen des Modells in Aufgabe AG6 erläutert. Wir stellen fest, dass diese Interpretationen *gleich* in dem von uns anfänglich in Abschnitt 8.3.3 erwähnten Sinne sind. Auch der Zusammenhang zwischen Problem- und Lösungsdomäne wurde von den Probanden im Rahmen der Aufgabe OF6b festgestellt und richtig interpretiert.

Die Sinnhaftigkeit der Trennung von Software- und Geschäftsanteilen wurde in Frage SF8 im Kontext des Requirements Engineerings in frühen Phasen mit durchschnittlich 6,6 Punkten bewertet. Die eigene Präzisierung der Vorstellung durch die Trennung wurde in Frage SF9 mit 6,8 Punkten bewertet. Die Probanden wurden gebeten, die Eignung des Basismodells zur Trennung des vorher genannten, im Vergleich zu UML-Klassendiagrammen, zu bewerten. Dies geschah durch Frage SF10 mit durchschnittlich 6,5 Punkten.

Abschließend wurden die Probanden gebeten zu bewerten, ob die Modellierung des Basismodells die Vorstellung über die Domäne präzisiert hat, dies geschah durch Frage SF13 mit durchschnittlich 7,6 Punkten. Aufbauend auf diese Frage wurden die Probanden gebeten, die Kommunizierbarkeit dieser Vorstellung innerhalb des Projektes zu bewerten. Dies geschah durch Frage SF14 mit durchschnittlich 6,6 Punkten.

► **K4.1: Die Artefakte, die gleiche Ausschnitte eines Problems bzw. einer Lösung mit Hilfe der gleichen Modellierungstechnik beschreiben, werden von unterschiedlichen Personen ähnlich modelliert.**

Im Rahmen von Aufgabe AG11 haben die Probanden ein eigenes Basismodell erstellt. Sieben Probanden haben alle relevanten Entitäten identifiziert und den richtigen Konzepten zugeordnet. In einem Modell wurden keine Stereotypen angeführt aber die richtigen Entitäten identifiziert.

Im Rahmen der Abbildung eines Sachverhaltes in Aufgabe AG11 wurden die

Probanden anschließend gebeten in Frage SF11b zu bewerten, ob sich alle relevanten Sachverhalte abbilden ließen. Dies wurde mit durchschnittlich 7,1 Punkten bewertet. Die Probanden wurden gebeten den Abbildevorgang auch im Vergleich zu UML-Klassendiagrammen in Hinblick auf Zielorientiertheit in Frage SF12 zu bewerten. Dies wurde mit durchschnittlich sieben Punkten bewertet.

- ▶ **K7.1: Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.**

Die Angebrachtheit der Repräsentation des Sachverhaltes beschrieben durch das Basismodell in Frage AG2 wurde durch Frage SF4 mit durchschnittlich 6,5 Punkte beurteilt. Dieses Basismodell wurde dann nochmals auf seine Angebrachtheit zur besseren Eignung zur Kommunikation innerhalb der Stakeholder im Vergleich zu UML-Klassendiagrammen in Frage AG6 mit durchschnittlich 6,8 Punkten bewertet.

8.3.7 Validität des Experiments

Konstruktvalidität. Das Design des Experiments folgt den Forschungszielen. Einerseits wurde mit der beschriebenen Durchführung sichergestellt, dass die Probanden im ersten Teil des Experimentes den Modellierungsansatz noch nicht kannten. Dies wurde mit Fragen sichergestellt, die dies überprüfen sollen.

Nach jeder Frage, bis auf JNF1, gab es die Möglichkeit Anmerkungen zu machen. Die Probanden wurden darauf hingewiesen, dass sie bei unklaren Fragen dies in den Anmerkungen vermerken sollen.

Die gekürzten Ergebnisse werden in Abschnitt 8.3.5 offen gelegt. Unsere, daraus gezogenen, Schlussfolgerungen sollten daher nachvollziehbar sein.

Interne Validität. Das Design des Experiments, zumindest was die geschlossenen Fragen angeht und deren Beantwortung, kann durch die wissenschaftliche Gemeinschaft nachvollzogen werden.

Die dargestellte Modell-Interpretation ist ein Ausschnitt aus den Antworten, die im Rahmen des Experimentes gegeben wurden. Sie wurden vom Autor selektiert, um bestimmte Sachverhalte besserhervorzuheben. Dies geschah nach bestem Wissen und Gewissen.

Die von den Probanden erstellten Modelle sind ausschnittsweise in den Ergebnissen repräsentiert, die Feststellung der Ähnlichkeit erfolgte nach dargelegten Gesichtspunkten und sollte reproduzierbar sein.

Externe Validität. Die durchgeführten Experimente in Form von Aufgaben und deren Auswertung sind nicht übertragbar auf ein industrielles Projekt. Hierfür sind die Modelle zu klein und die Problemstellung zu sehr auf zu untersuchende Aspekte getrimmt. Der Aspekt des Vorgehens wurde im Experiment vernachlässigt. Die Anwendbarkeit und Übertragbarkeit in reale Projekte wurde in Abschnitt 8.1 untersucht.

Die untersuchten Aspekte, z. B. Modellinterpretation und kanonische Modellbildung sollten zeigen, dass der Modellierungsansatz diese Aspekte berücksichtigt. Der zweite Aspekt ist für ein Projekt, in dem der Modellierungsansatz angewendet wird, nicht relevant, da das Basismodell in einer kollaborativen Tätigkeit erstellt wird und nicht mehrere Basismodelle parallel entstehen. Die Untersuchung dieses Aspekts dient dazu herauszufinden, ob Basismodelle von verschiedenen Modellierern gleich erstellt werden, eben außerhalb des Projekt-Kontextes. Dies sollte uns dabei helfen herauszufinden, ob die Modelle in der Praxis, wenn sie gleiche Sachverhalte beschreiben, auch eine ähnliche Strukturierung aufweisen werden.

Zuverlässigkeit. Wir haben in Abschnitt 8.3.5 alle relevanten Informationen dargestellt, die im Rahmen des Experiments erhoben wurden. Teilweise wurden diese gekürzt und zur besseren Darstellung von Aspekten hervorgehoben.

Die Ergebnisse der Modellinterpretation und der Modellbildung durch die Probanden mussten durch den Autor so zusammengestellt werden, dass sie im Rahmen der Ergebnispräsentation angebracht dargestellt werden konnten. Wir haben nach bestem Wissen und Gewissen versucht, diese Darstellung so neutral wie möglich zu wählen und gehen davon aus, dass eine dritte Person auf ähnliche Ergebnisse kommen wird.

8.4 Diskussion der Analyse-Ergebnisse

In diesem Abschnitt diskutieren wir die Ergebnisse der Analyse aus den Abschnitten 8.1.7, 8.2.5 und 8.3.6.

Wir betrachten hier aus den Analyse-Kapiteln jeweils einen Teil der Beiträge zur Evaluierung. Hier sei darauf hingewiesen, dass es im Rahmen der Arbeit nicht möglich war, alle diese durch die Kriterien ausgedrückten Optimalzustände nachweislich zu erreichen. Wir sehen diese Arbeit als Beitrag, sich den Optimalzuständen zu nähern.

Die Bewertung der evaluierten Eigenschaften fand teilweise mit Hilfe einer achtstufigen Skala statt. Die möglichen Antworten lagen zwischen *trifft völlig zu* und *trifft überhaupt nicht zu*. Wir sprechen von acht Punkten, wenn wir von einer völlig zutreffenden Aussage sprechen und von einem Punkt, wenn wir von einer Aussage sprechen, die überhaupt nicht zu trifft. Die achtstufige Skala erlaubt es uns, keine neutralen Aussagen treffen zu müssen. Alles was über oder gleich fünf Punkten liegt, trifft daher eher zu und alles was unter oder gleich vier Punkten liegt, trifft eher nicht zu.

► K1.1: Die Interpretation eines Modells durch verschiedene Personen führt zu ähnlichen Ergebnissen.

Im Laborexperiment in Abschnitt 8.3 haben wir die Interpretation eines Basismodells durch unterschiedliche Probanden untersucht. Die Probanden haben in ihren eigenen Worten beschrieben, was das Basismodell beschreibt und welche Bedeutung die Beziehungen im Basismodell haben. Das Experiment zeigte, dass auf der gewählten Abstraktionsebene des Experiments ein Basismodell gleich interpretiert wird. Wobei die Gleichheit entsprechend der in Abschnitt

8.3.4 erläuterten Weise betrachtet wurde.

Zusätzlich haben wir im Laborexperiment untersucht, ob die Probanden erkennen, dass im Basismodell Problem- und Lösungsdomäne dargestellt werden. Um dies festzustellen, haben wir um eine Interpretation der Beziehungen zwischen Entitäten der Problem- und Lösungsdomäne gebeten. Dieser Zusammenhang wurde von allen Probanden festgestellt.

Weitere Einflussfaktoren auf die Reduktion von inhaltlichen Missverständnissen wurden im Laborexperiment und in den Fallstudien auf ihre Erfüllung hin evaluiert. So wurden im Laborexperiment die Probanden gebeten zu bewerten, ob die Modellierung des Basismodells ihnen dabei geholfen hat, die Vorstellung über die darzustellende Domäne zu präzisieren. Dies wurde mit durchschnittlich 7,6 Punkten bestätigt.

Die Eindeutigkeit und Verständlichkeit eines Basismodells ist wichtiger Bestandteil um inhaltliche Missverständnisse zu reduzieren. Wir haben im Rahmen der Befragung einen domänenfernen Experten gebeten, das Basismodell auf diese Kriterien hin zu bewerten, dies wurde mit sieben und sechs Punkten bewertet. Die Verständlichkeit wurde zwar noch gut bewertet, ist aber deswegen nicht hervorragend, da der Experte die Domäne nicht kannte, aber durch die Konzepte und Beziehungen, die im Basismodell als Stereotypen verwendet wurden, zumindest zu erkennen war, welche Bedeutung die Entitäten für die dargestellte Domäne haben. Diese Erläuterung wird auch dadurch bestätigt, dass der Beitrag zur Klärung von inhaltlichen Missverständnissen durch das Basismodell nur mit fünf Punkten bewertet wurde. Wir sehen als Ursache des Ergebnisses die Tatsache, dass die Begrifflichkeiten nur durch die Stereotypisierung und den Beziehungen in ihrer Bedeutung erläutert werden. Wenn die gewählten Begrifflichkeiten unklar sind, ist zwar deren grobe Bedeutung durch die Stereotypisierung erkennbar, aber Details müssen unabhängig davon kommuniziert werden.

- **K2.1: Die Artefakte stehen in einer Beziehung zueinander, sodass keine Interpretationslücken geschlossen werden müssen.**

Ein Beitrag zur Reduktion von Interpretationslücken ist die Identifikation von Inhalten auf Basis des Basismodells zur Detaillierung in anderen Artefakt-Typen. Durch diese Vorgehensweise stellen wir sicher, dass die Inhalte in anderen Artefakten einen Bezug zum Basismodell haben. Im Rahmen der Industrie- und Praktikumsstudie wurde bewertet, ob das Basismodell sich als Basis für dieses Vorgehen eignet. Dies geschah mit acht und durchschnittlich 7,7 Punkten. Diese guten Ergebnissen führen wir darauf zurück, dass das Basismodell als Produkt eines gemeinsamen Konstruktionsprozesses auch die Vorstellung der an der Konstruktion beteiligten Personen repräsentiert und daher auch die Verknüpfung zu anderen Artefakten für diese Gruppe einfach möglich war.

In der Industrie- und Praktikumsstudie haben wir auch erhoben, ob das Basismodell zur Klärung von Abhängigkeiten zwischen den Artefakten geeignet ist. Dies wurde mit acht und durchschnittlich 7,7 Punkten bestätigt. In den Studien wurde das Basismodell als eher abstraktes Artefakt verwendet, zu dem alle weiteren Artefakte konform sein mussten. Diese Konformität ist dann gegeben, wenn die in weiteren Artefakten detaillierteren Inhalten in keinem Wi-

derspruch zum Basismodell standen. Widersprüche zwischen den Artefakten konnten so zwar nicht direkt erkannt werden, durch die Sicherstellung der Konformität zum Basismodell war es aber möglich, die Artefakte auf einer hohen Abstraktionsebene konsistent zu halten.

- ▶ **K3.1: Die Artefakte, erstellt in frühen Phasen des Requirements Engineerings, stehen in keinem Widerspruch untereinander.**

Wie im vorhergehenden Kriterium erläutert wurde, wurde das Basismodell u.a. als Basis für weitere Artefakte verwendet. Im Rahmen der Industrie- und Praktikumsstudie haben wir erhoben, ob die Verwendung des Basismodells zu konsistenten Artefakten führt. Dies wurde mit sieben und durchschnittlich 7,7 Punkten bestätigt.

Wir sehen die Struktur der Inhalte von Artefakten als wichtigen Einflussfaktor auf die Feststellung der Konsistenz von Artefakten. Im Rahmen der Industrie- und Praktikumsstudie haben wir erhoben, ob die Verwendung des Basismodells zu besser strukturierten Artefakten führt. Dies wurde mit sechs und durchschnittlich 7,7 Punkten bewertet. Diese zwei Bewertungen weichen voneinander ab, wir erklären uns die bessere Bewertung in der Praktikumsstudie dadurch, dass die Anzahl der Artefakte und die darin explizierten Sachverhalte überschaubarer waren, als sie es im Rahmen des Industrieprojektes waren.

- ▶ **K4.1: Die Artefakte, die gleiche Ausschnitte eines Problems bzw. einer Lösung mit Hilfe der gleichen Modellierungstechnik beschreiben, werden von unterschiedlichen Personen ähnlich modelliert.**

Im Rahmen des Laborexperimentes haben wir untersucht, ob durch die Vorgehensweise auch unterschiedliche Personen eine bestimmte Domäne so beschreiben, dass ähnliche Basismodelle entstehen. Die Ergebnisse haben gezeigt, dass dies der Fall ist. Hier sei kritisch angemerkt, dass dies auch deshalb der Fall sein kann, weil die beschriebene Domäne ein *Toy-Example* war.

Die Vorgaben, der durch die Modellierungssprache ausdrückbaren Inhalte, reduzieren die möglichen damit zu modellierenden Domänen. Wir haben im Rahmen des Experimentes daher auch erfragt, ob sich alle subjektiv relevanten Sachverhalte abbilden ließen. Dies wurde durchschnittlich mit 7,1 Punkten bestätigt. Auch hier sei angemerkt, dass dieses Ergebnis auch daran liegen kann, dass die im Experiment beschriebene Domäne eine Beschreibung eines Sachverhaltes war, der für die Zielgruppe der Arbeit relevant ist (prozessorientierte Informationssysteme).

Wir haben die Probanden im Rahmen des Laborexperimentes auch gebeten zu bewerten, inwieweit die Modellierungssprache besser geeignet ist, als die in diesem Kontext häufig eingesetzten UML-Klassendiagramme, um die beschriebene Domäne (inklusive Trennung der Problem- und Lösungsdomäne) abzubilden. Dies wurde mit durchschnittlich 6,5 bewertet. Das noch positive Ergebnis zeigt, dass dies den Probanden subjektiv so vorgekommen ist. Auch hier gehen wir davon aus, dass dies daran liegt, dass es für die zu modellierenden Sachverhalte der Zielgruppe Konzepte und Beziehungen gibt, die eine Abbildung erleichtern.

Im Rahmen der Praktikumsstudie hatten wir die Möglichkeit, die entstandenen

Artefakte zu analysieren. Wir haben die Verknüpfung von den Basismodellen zu anderen Artefakten betrachtet. Um einen Beitrag zur Erfüllung des Kriteriums feststellen zu können, haben wir einen Artefakt-Typ betrachtet, der von allen Teams eingesetzt wurde, das Datenmodell. Wir haben festgestellt, dass die Strukturen im Basismodell 1:1 als Entitäten in den Datenmodellen abgebildet werden. Wir sehen dies als Indiz dafür, dass die Verwendung des Basismodells im Entwicklungsprozess die Modellierung der Inhalte in anderen Artefakten lenkt.

- ▶ **K5.1: Alle Ausschnitte eines Gesamtzusammenhangs werden von zumindest einem Artefakt beschrieben.**

Artefakte in frühen Phasen des Requirements Engineerings werden unabhängig voneinander erstellt. Dies und unterschiedliche Prioritäten bei der Modellbildung führen dazu, dass die in Artefakten erhobenen Inhalte häufig unausgeglichen repräsentiert sind. Das Basismodell repräsentiert ein Gesamtverständnis auf einer hohen Abstraktionsebene. Das Basismodell reicht nicht aus, um die in späteren Phasen des Softwareentwicklungsprozesses notwendigen Details zu repräsentieren. Wir sehen daher das Wissen dann als ausgeglichen repräsentiert, wenn jeder Ausschnitt aus dem Basismodell von zumindest einem weiteren Artefakt beschrieben und typischerweise detailliert wurde. Ein Ausschnitt in diesem Kontext stellt zumindest einen Zusammenhang dar. Der Zusammenhang stellt eine Beziehung zwischen zumindest zwei Sachverhalten her [Dud11]. Ein Sachverhalt im Basismodell entspricht einer Entität. Ein Ausschnitt besteht daher zumindest aus zwei Entitäten und einer stereotypisierten Beziehung zwischen diesen Entitäten. Im Rahmen der Industrie- und Praktikumsstudie wurde bewertet, ob das Basismodell dabei hilft, Wissen in anderen Artefakten ausgeglichen zu erheben, dies wurde mit sieben und durchschnittlich 7,5 Punkten bewertet.

Wir sehen das Ergebnis als Indiz dafür, dass ein Basismodell dabei helfen kann, das zu detaillierende Wissen zu identifizieren. Eine zusätzliche Gefahr entsteht im Softwareentwicklungsprozess dann, wenn das Basismodell nicht vollständig ist (siehe dazu Abschnitt 6.4). Weiter kann durch die hohe Abstraktionsebene eines Basismodells nicht sicher gestellt werden, dass es innerhalb eines Ausschnittes nicht noch mehr Details gibt, die wichtig wären und noch weiter detailliert werden sollten.

- ▶ **K6.1: Eine Verknüpfung zu anderen Artefakten ist angemessen möglich.**

Wir sehen die Verbesserung der inneren Struktur von Artefakten, die auf dem Basismodell basieren, als Indiz dafür, dass die Verknüpfung einen Mehrwert bringt. Dies wurde im Rahmen der Industrie- und Praktikumsstudie erhoben und mit sechs und durchschnittlich 7,7 Punkten bewertet. Weiteres Indiz ist die Verbesserung der Qualität der Artefakte, die auf dem Basismodell basieren, dies wurde mit sieben und durchschnittlich 7,5 Punkten bestätigt.

Ein Proband führte an, dass das Basismodell die Verbindung zu anderen Artefakten sehr gut herstellt.

- ▶ **K7.1: Es existiert ein angemessenes stakeholderweites Verständnis über die Problem- und Lösungsdomäne.**

Das stakeholderweite und angemessene Verständnis über die Problem- und Lösungsdomäne konnten wir nicht messen. Wir diskutieren in diesem Punkt daher Indizien, die darauf hinweisen, dass wir mit dem Modellierungsansatz zumindest einen Beitrag leisten, dieses zu entwickeln.

Um ein stakeholderweites Verständnis repräsentieren zu können, müssen die in einem Modell explizierten Sachverhalte auch kommunizierbar sein.

Im Rahmen des Laborexperimentes haben die Probanden bewertet, ob sich das Basismodell zur Repräsentation der Sachverhalte eignet. Dies wurde mit durchschnittlich 6,5 Punkten bewertet.

Ein domänenferner Experte wurde gebeten zu bewerten, ob sich ein Basismodell dazu eignet, ein Gesamtverständnis zu repräsentieren. Dies wurde mit acht Punkten bestätigt. Einschränkend sei hier noch angemerkt, dass der domänenferne Experte aber bei der Bewertung, ob das Basismodell zur Klärung von inhaltlichen Missverständnissen geeignet ist, dies nur mit fünf Punkten bewertet hat. Wir erklären uns diese Diskrepanz damit, dass für den Zweck der Repräsentation eines Gesamtverständnisses absichtlich auf Details verzichtet wird, diese Details aber später im Entwicklungsprozess notwendigerweise expliziert werden müssen, sich dafür aber das Basismodell nicht mehr eignet. Dafür haben wir im Modellierungsansatz vorgesehen, dass andere Artefakt-Typen zur Detaillierung verwendet werden.

Ein Kommentar eines Probanden sagt aus, dass der Modellierungsansatz zu einem guten *Rich-Picture* und zur besseren Repräsentation des Umfelds führt.

Im Rahmen der Industrie- und Praktikumsstudie haben wir erhoben, ob die Gesamtvorstellung mit dem Basismodell klar geworden ist. Dies wurde mit acht und durchschnittlich 7,3 Punkten bestätigt.

Einige Kommentare geben einen zusätzlichen Hinweis darauf, dass wir einen Beitrag zur Schaffung eines stakeholderweiten Verständnisses geschaffen haben. So ist z. B. von der Schaffung eines sehr guten *Rich-Pictures*, dem Abdecken mehrerer Modelle durch das Basismodell sowie der Schaffung einer genaueren Vorstellung über die Use-Cases, usw. einer Domäne, die Sprache.

Ein Proband führte an, dass die Darstellung der Problemwelt und Lösungswelt auf den ersten Blick zu ähnlichen Aussagen führt. Wir sehen dies als Gefahr, da wir die Darstellung der Problem- und Lösungsdomäne in einem Modell als eine wichtige Facette des Modellierungsansatzes sehen. Ein anderer Kommentar trifft eine ähnliche Aussage: „[...] Auch wenn ich mich überwinden musste die Entitäten auf den ersten Blick redundant zu modellieren, so macht diese Trennung durchaus Sinn.“ Wir sehen dies als Hinweis darauf, dass wir die Motivation hinter der Trennung und gleichzeitigen Darstellung im Vorgehen besser kommunizieren müssen.

Anwendbarkeit. Im Rahmen der Evaluierung haben wir Fragen zur Anwendbarkeit der Methode sowie dem Produkt, dem Basismodell, gestellt. U.a. haben wir im Rahmen der Industrie- und Praktikumsstudie erhoben, ob der Modellierungsansatz für den Einsatz in frühen Phasen des Requirements Engineerings geeignet ist, dies wurde mit sieben und durchschnittlich 7,5 Punkten bestätigt. Ein weiterer Aspekt

der Anwendbarkeit ist die Einfachheit, Strukturiertheit und Eindeutigkeit des Vorgehens, diese wurden mit sieben, acht und sieben sowie durchschnittlich 6,7, 7,5 und 7,5 Punkten bewertet.

Ein weiteres Indiz auf die Anwendbarkeit ist durch ein Kommentar gegeben. Im Kommentar ist von einem eindeutigen Vorgehen bei der Erstellung des Basismodells die Rede.

Im Rahmen der Praktikumsstudie haben wir festgestellt, dass die Kompetenzfragen eher als Fragen gesehen wurden, die dabei helfen, keine Aspekte der zu erfassenden Domäne zu vergessen. Die ontologische Basis war für die Studienteilnehmer wichtiger als die Vorgehensweise.

Die Fehleranfälligkeit des Modellierungsansatzes wurde im Rahmen der Industrie- und Praktikumsstudie mit sechs und durchschnittlich 6,8 Punkten bewertet. In einem Kommentar wird darauf hingewiesen, dass der Modellierungsansatz durchaus Einarbeitszeit beansprucht. Wir sehen die Einfachheit des Modellierungsansatzes in einem Spannungsfeld mit den Vorgaben im Konstruktionsprozess. Je mehr Vorgaben durch die einzelnen Schritten und die Kompetenzfragen eingeführt werden, desto größer wird der Bedarf den Konstruktionsprozess anpassbar zu machen, da dieser sonst zu komplex wird um in frühen Phasen des Requirements Engineerings noch eingesetzt zu werden.

Mehrwert. Im Rahmen der Evaluierung haben wir in den Industrie- und Praktikumsstudien auch gefragt, ob der Modellierungsansatz besser geeignet ist, als der zuvor verwendete. Dies wurde mit acht und acht Punkten bestätigt. Dieses Ergebnis ist eindeutig ausgefallen. Wir gehen davon aus, dass diese Eindeutigkeit dadurch entstanden ist, dass weder von den Probanden der Industriestudie noch von den Probanden der Praktikumsstudien zuvor eine strukturierte Vorgehensweise in frühen Phasen des Requirements Engineerings angewendet wurde und dadurch womöglich jeder Modellierungsansatz, der einen erkennbaren Mehrwert für die Probanden bietet, so eine positive Bewertung erhalten hätte.

Als ein wichtiges Indiz für den Mehrwert sehen wir die Bewertung durch die Frage im Rahmen der Industrie- und Praktikumsstudie, ob die Produktivität in frühen Phasen des Requirements Engineerings durch die Verwendung des Modellierungsansatzes gefördert wurde, dies wurde mit sieben und durchschnittlich 7,3 Punkten bewertet.

Einen weiteren Hinweis auf einen Mehrwert durch den Einsatz des Modellierungsansatzes geben die Kommentare, dass (a) der Modellierungsansatz zu einer früheren intensiven Diskussion geführt hat und (b), dass das Basismodell als *Leitmodell* sehr hilfreich im Requirements Engineering war. Ein weiterer Kommentar spricht direkt einen definitiven Mehrwert bezüglich der bisherigen UML-Modellierung an.

Zur Anzahl von Entitäten im Basismodell. Mehrfach wurde im Rahmen der Evaluation von verschiedenen Probanden (im Rahmen der Befragung eines domänenfernen Experten, der Industrie- und Praktikumsstudie) angeführt, dass die Anzahl der Entitäten und Beziehungen in einem Basismodell sehr schnell mehr werden. Auch wurde in Kommentaren die Notwendigkeit einer Werkzeugunterstützung an-

geführt. Die Werkzeugunterstützung wird auch durch die schnell steigende Anzahl von Entitäten im Basismodell motiviert.

Wir sehen dies auch als Begründung für die Bewertung auf die Frage, ob der Modellierungsansatz minimalistisch ist. Dies wurde im Rahmen der Industriestudie mit fünf Punkten bewertet.

Wir sehen dies als Gefahr für das Basismodell, da das Ziel, ein Gesamtverständnis auf einer hohen Abstraktionsebene zu repräsentieren, dann gefährdet ist, wenn die quantitative Komplexität des Basismodells zunimmt. Eine mögliche Lösung des Problems ist die Einführung eines Abstraktionskonzeptes, das die im Basismodell modellierten Sachverhalte auf unterschiedlichen Abstraktionsebenen repräsentierbar macht.

8.5 Zusammenfassung

In diesem Kapitel haben wir die Evaluierung des Modellierungsansatzes vorgenommen. Hierzu wurden in Abschnitt 8.1 die Ergebnisse und Analyse der durchgeführten Industrie- und Praktikumsstudie präsentiert. Aufbauend auf die Industriestudie wurde ein Teil des entstandenen Basismodells in Abschnitt 8.2 von einem domänenfernen Experten bewertet. Einige Facetten der Problemstellung konnten wir nicht im Rahmen der Fallstudien evaluieren. Zum Schließen der Lücken wurde ein Laborexperiment durchgeführt, das wir in Abschnitt 8.3 präsentierten. In Abschnitt 8.4 wurden die wichtigsten Ergebnisse der Analysen abschließend diskutiert.

Zusammenfassung, Ausblick und persönliches Resümee

In dieser Arbeit haben wir einen Ansatz vorgestellt, der in frühen Phasen des Requirements Engineerings eingesetzt werden kann, um ein Gesamtverständnis¹ zu erarbeiten. Dazu wurde in der Praxis erhoben, welche Artefakt-Typen in frühen Phasen erstellt werden und welche Probleme bei der Verwendung der Artefakte im Requirements Engineerings und darüber hinaus auftreten, siehe dazu Kapitel 3. Einige der erhobenen Probleme konnten mit der Analyse von den in Artefakten erfassten Inhalten in Kapitel 4 bestätigt werden. In Kapitel 5 haben wir die erhobenen Probleme mit deren Auswirkungen analysiert und einen Lösungsvorschlag erarbeitet. Aufbauend auf die Lösungsidee haben wir in Kapitel 6 eine ontologische Basis vorgeschlagen. Die ontologische Basis dient als Terminologie einer Sprache zur Beschreibung eines Gesamtverständnisses. Die Vorgehensweise zur Erhebung dieses Gesamtverständnisses haben wir in Kapitel 7 vorgestellt. In diesem Kapitel zeigen wir auch beispielhaft die Verknüpfung zu anderen Artefakt-Typen. In Kapitel 8 haben wir den Modellierungsansatz in verschiedenen Softwareprojekten angewendet und evaluiert. Die offenen Kriterien, die nicht durch die Fallstudien evaluiert werden konnten, wurden im Rahmen eines Laborexperimentes evaluiert.

9.1 Ausblick

Dieser Abschnitt beschreibt zukünftige Arbeiten, die aufbauend auf die Ergebnisse, expliziert in dieser Arbeit, durchgeführt werden können.

9.1.1 Ontologische Basis und Vorgehen für andere Domänen

Im Rahmen der Arbeiten am Lehrstuhl von Prof. Broy an verschiedenen Projekten, kam der Autor häufig in Berührung mit eingebetteten Systemen. Der Anwendungsbereich eines eingebetteten Systems ist anders, als er im Rahmen eines operativen

¹ In Kontext dieser Arbeit ist das Gesamtverständnis das gemeinsame Verständnis aller relevanten Stakeholder über die Problem- und Lösungsdomäne auf einer gemeinsam festgelegten Abstraktionsebene. Die Definition befindet sich auf Seite 4.

Informationssystem ist, das in einem Umfeld betrieben wird, das vor allem an Geschäftsprozessen ausgerichtet ist. Die Idee hinter den in dieser Arbeit vorgestellten Modellierungsansatz auf bestimmte (eventuell auch eingeschränkte) Facetten des Anwendungsbereiches von eingebetteten Systemen zu übertragen, stellt einen herausfordernden nächsten Schritt dar, der aufbauend auf diese Arbeit in Betracht gezogen werden kann.

9.1.2 Anforderungsspezifikation

Im Rahmen der Verwendung von Domänenwissen im Requirements Engineering wurde vom Autor eine Bachelorarbeit betreut, die das Wissen einer Problemdomäne als Basis für das Formulieren von Anforderungen in (beinahe) natürlicher Sprache ermöglichte [Has11]. Dazu wurde eine Menge von Satzmustern erarbeitet, die bei Instanziierung Wissen über die Problemdomäne beinhalten konnten bzw. das Wissen über die Problemdomäne teilweise erweiterten. Das mit Satzmustern beschriebene Wissen inkludierte bereits Wissen über das System (daher der Lösungsdomäne) und deren Zusammenhang mit dem Wissen über die Problemdomäne. Die Weiterverwendung von Wissen erarbeitet durch den Ansatz in dieser Dissertation im Requirements Engineering wäre ein wichtiger Schritt um den Übergang zu (semi-)formalen Anforderungen zu ermöglichen.

9.1.3 Unterstützung bei der Auswahl von Entitäten für die Verknüpfungen

Das als Produkt des Modellierungsansatzes erstellte Basismodell dient als Ausgangsbasis für weitere Artefakte, erstellt durch andere Modellierungstechniken. Die in dieser Arbeit vorgestellten Verknüpfungen sind beispielhaft und müssen nicht 1:1 befolgt werden. Ein nächster Schritt nach dieser Arbeit könnte die Erarbeitung eines Regelkatalogs sein, der für bestimmte Modellierungstechniken vorgibt, wie eine Auswahl von Entitäten zur Detaillierung der Auswahl in der gewählten Technik aussehen kann. Wir haben in dieser Arbeit versucht, die Idee bei der Verknüpfung beispielhaft zu beschreiben. Eine präzisere Beschreibung, sowie die Ausarbeitung von Regeln, wäre für weitere Arbeiten ein möglicher Schritt.

9.1.4 Werkzeugunterstützung

Modellierung in frühen Phasen des Requirements Engineerings wird häufig auch manuell und ohne Werkzeugunterstützung durchgeführt. Wenn eine Werkzeugunterstützung verwendet wird, sind die damit erstellten Artefakte voneinander losgelöst.

Im Rahmen der Evaluierung des Modellierungsansatzes wurde von mehreren Personen erwähnt, dass eine Werkzeugunterstützung notwendig ist, da die Anzahl der Entitäten und daher die quantitative Komplexität² der Basismodelle schnell größer

² Die quantitative Komplexität beschreibt die Komplexität eines Modells durch die Menge von modellierten Modellelementen und deren Beziehungen [AF06].

werden kann. Um die Diffusion in die Industrie zu ermöglichen, ist eine angemessene Werkzeugunterstützung eine der Voraussetzungen und wäre ein logischer Schritt für die nächsten Arbeiten.

9.2 Persönliches Resümee

Für den Autor dieser Arbeit war die Modellierung von Wissen über die Problem- und Lösungsdomäne bereits vor dieser Arbeit ein wichtiges Thema. Getrieben durch die Erfahrung in der Praxis wurde das Thema aufgegriffen, um (a) den Begriff für eine bestimmte Klasse von Problemen zu schärfen und (b) zu zeigen, wie eine hilfreicher Modellierungsansatz aussehen kann. Dies ist umso wichtiger, da es in der Literatur keinen Ansatz gibt, der speziell für den Einsatz in frühen Phasen des Requirements Engineerings erstellt wurde.

Ein Problem, auf das der Autor in der Praxis häufig stieß, ist das Fehlen eines Übersichtswissen (das wir in dieser Arbeit auch Gesamtverständnis nennen), das den großen Zusammenhang einer Domäne abbildet. Der Beitrag eines Entwicklers an einem Projekt ist häufig sehr isoliert, teilweise wurde von Industriepartnern in der Zusammenarbeit angeführt, dass der Überblick vollständig fehlt, dies wird auch in der Literatur, z.B. von [CKI88, Sel12] bestätigt. Es wird daher an Softwareentwicklern an Softwareteilen gearbeitet, von denen nicht bekannt ist, welche Rolle sie im Softwareprojekt spielen. Das Fehlen des Überblickswissen führte zu einer eigenen Vorstellung des Softwareentwicklers über die Funktionsweise des Softwareteils im System.

Die Modellierung der Problem- und Lösungsdomäne in frühen Phasen des Requirements Engineerings ist für den Autor ein sehr wichtiges und auch zukunftssträchtiges Thema, das (womöglich) wegen seiner nicht einfach greifbaren Problematik und Komplexität, in der Wissenschaft nicht präzisiert, aber durchaus erwähnt wird. Wir sehen diese Arbeit als einen Anfang, der zeigt, wie das Gesamtverständnis in frühen Phasen des Requirements Engineerings für eine bestimmte Domäne erarbeitet und eingesetzt werden kann.

Literaturverzeichnis

- [ABC⁺02] ACKERMANN, Jörg ; BRINKOP, Frank ; CONRAD, Stefan ; FETTKE, Peter ; FRICK, Andreas ; GLISTAU, Elke ; JAEKEL, Holger ; KOTLAR, Otto ; LOOS, Peter ; MRECH, Heike ; ORTNER, Erich ; RAAPE, Ulrich ; OVERHAGE, Sven ; SAHM, Stephan ; SCHMIETENDORF, Andreas ; TESCHKE, Thorsten ; TUROWSKI, Klaus: *Vereinheitlichte Spezifikation von Fachkomponenten: Memorandum des Arbeitskreises 5.10.3 Komponentenorientierte betriebliche Anwendungssysteme*. Gesellschaft für Informatik, 2002
- [AC04] AITCHISON, Jean ; CLARKE, Stella D.: *The thesaurus: a historical viewpoint, with a look to the future*. Haworth Press,, 2004
- [AF06] AMANN, Kathrin ; FLEISCHMANN, Andreas: Bewertung der Verständlichkeit graphischer Modelle. In: *Modellierung*, 2006, S. 281–284
- [AGP04] ASUNCIÓN GÓMEZ-PÉREZ, Oscar C. Mariano Fernández-López: *Ontological Engineering*. Springer, 2004
- [All09] ALLWEYER, Thomas: *BPMN 2.0 Business Process Model and Notation. Einführung in den Standard für die Geschäftsprozessmodellierung*. Books on Demand, 2009
- [Als58] ALSTON, William P.: Ontological commitments. In: *Philosophical Studies* 9 (1958), S. 8–17. – ISSN 0031–8116. – 10.1007/BF00797866
- [AS03] ANGELA SANDMANN, Gunter L.: Lernstrategien und Domänenwissen. In: *Zeitschrift für Psychologie* (2003), S. 171–192
- [AW05] AURUM, Aybüke ; WOHLIN, Claes: *Engineering and Managing Software Requirements*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2005. – ISBN 3540250433
- [BADW04] BREWSTER, Christopher ; ALANI, Harith ; DASMAHAPATRA, Srinandan ; WILKS, Yorick: Data Driven Ontology Evaluation. In: *International Conference on Language Resources and Evaluation*, 2004
- [Ber05] BERTRAM, Jutta: *Einführung in die inhaltliche Erschließung: Grundlagen – Methoden – Instrumente*. Ergon, 2005
- [Bey03] *Kapitel Prozessanalyse*. In: BEYER, H.-T.: *Allgemeine Betriebswirtschaftslehre*. Online: <http://www.economics.phil.uni-erlangen.de/lehre/bwl-archiv/lehrbuch>, 2003
- [BGM05] BRANK, Janez ; GROBELNIK, Marko ; MLADENIĆ, Dunja: A survey of ontology evaluation techniques. (2005)

- [BHKN03] BECKER, Jörg ; HOLTEN, Roland ; KNACKSTEDT, Ralf ; NIEHAVES, Björn: Forschungsmethodische Positionierung in der Wirtschaftsinformatik – epistemologische, ontologische und linguistische Leitfragen – / Westfälische Wilhelms-Universität Münster – Institut für Wirtschaftsinformatik. 2003. – Forschungsbericht
- [Bjø12] BJØRNER, Dines: *Towards a Theory of Domain Descriptions*. 2012
- [BKR08] BECKER, Jörg ; KUGELER, Martin ; ROSEMAN, Michael: *Prozessmanagement: Ein Leitfaden zur prozessorientierten Organisationsgestaltung*. Springer, 2008
- [BNK04] *Kapitel Bezugrahmen zur epistemologischen Positionierung der Referenzmodellierung.* In: BECKER, Jörg ; NIEHAVES, Björn ; KNACKSTEDT, Ralf: *Referenzmodellierung: Grundlagen, Techniken und domänenbezogene Anwendung*. Physica-Verlag Heidelberg, 2004, S. 1–13
- [Bor97] BORST, Willem N.: *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Enschede, Diss., September 1997. <http://doc.utwente.nl/17864/>
- [BOS06] BOMQVIST, Eva ; ÖGREN, Annika ; SANDKUHL, Kurt: *Ontology Construction in an Enterprise Context: Comparing and Evaluating two Approaches*. (2006)
- [Bro02] BROCKHAUS: *Brockhaus Enzyklopädie 2002 digital*. F. A. Brockhaus AG, 2002
- [Bru10] BRUNOLD UND PARTNER UNTERNEHMENSBERATUNG: *Organisationsmodell*. <http://www.p3wiki.org/w/index.php?title=Organisationsmodell>. Version: März 2010
- [BS01] BROY, Manfred ; STØLEN, Ketil: *Specification and development of interactive systems: focus on streams, interfaces, and refinement*. Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2001. – ISBN 0–387–95073–7
- [BS03] BROY, M. ; STEINBRÜGGEN, R.: *Modellbildung in der Informatik*. Springer, 2003. – ISBN 9783540442929
- [CH01] COCKBURN, A. ; HIGHSMITH, J.: *Agile software development, the people factor*. In: *Computer* 34 (2001), Nr. 11, S. 131–133
- [CKI88] CURTIS, Bill ; KRASNER, Herb ; ISCOE, Neil: *A field study of the software design process for large systems*. In: *Commun. ACM* 31 (1988), November, Nr. 11, S. 1268–1287
- [CM02] COUGHLAN, Jane ; MACREDIE, Robert D.: *Effective Communication in Requirements Elicitation: A Comparison of Methodologies*. In: *Requirements Engineering* 7 (2002), Nr. 2, S. 47–60
- [Coc07] COCKBURN, Alistair: *Use Cases Effektiv Erstellen*. Hüthig Jehle Rehm, 2007
- [CY94] COAD, Peter ; YOURDON, Edward: *OOA: Objekt-Orientierte Analyse*. Prentice Hall, 1994
- [Dor11] *Kapitel Object-Process Methodology for Structure-Behavior Codesign*.

- In: DORI, Dov: *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*. Springer, 2011, S. 209–258
- [Dud11] DUDEN: *Duden: Deutsches Universalwörterbuch*. Dudenverlag, 2011
- [DW03] DEKKERS, Makx ; WEIBEL, Stuart: State of the Dublin Core Metadata Initiative. In: *D-Lib Magazine* 9 (2003)
- [e-t10] E-TEACHING.ORG: *Wissensmanagement*. <http://www.e-teaching.org/projekt/organisation/wissen/>, Dezember 2010
- [Ent93] ENTMAN, Robert M.: Framing: Toward Clarification of a Fractured Paradigm. In: *Journal of Communication* (1993), S. 51–58
- [EW05] EVERMANN, Joerg ; WAND, Yair: Ontology based object-oriented domain modelling: fundamental concepts. In: *Requirements Engineering* 10 (2005), S. 146–160. – ISSN 0947–3602. – 10.1007/s00766-004-0208-2
- [FE07] FANG, J. ; EVERMANN, J.: Evaluating Ontologies: Towards a Cognitive Measure of Quality. In: *EDOC Conference Workshop, 2007. EDOC '07. Eleventh International IEEE, 2007*, S. 109–116
- [Fen03] FENSEL, Dieter: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer, 2003
- [Fer11] FERNÁNDEZ, Daniel M.: *Requirements Engineering: Artefact-Based Customisation*, Technische Universität München, Diss., 2011
- [FH91] FLOR, Nick V. ; HUTCHINS, Edwin L.: Analyzing Distributed Cognition in Software Teams: A Case Study of Team Programming During Perfective Software Maintenance. In: *Empirical Studies of Programmers: Fourth Workshop*, 1991
- [FHL⁺98] FALKENBERG, Eckhard D. ; HESSE, Wolfgang ; LINDGREEN, Paul ; NILSSON, Björn E. ; OEI, J. L. H. ; ROLLAND, Colette ; STAMPER, Ronald K. ; ASSCHE, Frans J. M. V. ; VERRIJN-STUART, Alexander A. ; VOSS, Klaus: A Framework of Information System Concepts. In: *The FRISCO Report* (1998)
- [FK03] FOUCAULT, Michel ; KÖPPEN, Ulrich: *Die Ordnung der Dinge. Eine Archäologie der Humanwissenschaften*. Suhrkamp Verlag, 2003
- [FL99] FERNÁNDEZ-LÓPEZ, M.: Overview of methodologies for building ontologies. In: *Workshop Ontologies and Problem-Solving Methods: Lessons Learned and Future Trends de la conferencia International Joint Conference for Artificial Intelligence*, 1999
- [FL03] FETTKE, Peter ; LOOS, Peter: Ontologische Evaluierung des Semantischen Objektmodells. In: *MobIS*, 2003, S. 109–129
- [FLGPJ97] FERNÁNDEZ-LÓPEZ, M. ; GÓMEZ-PÉREZ, A. ; JURISTO, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In: *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, 1997
- [FLPW11] FERNANDEZ, D. M. ; LOCHMANN, K. ; PENZENSTADLER, B. ; WAGNER, S.: A case study on the application of an artefact-based requirements

- engineering approach. In: *15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)* (2011), S. 104–113(9)
- [FLW00] FAYAD, Mohamed E. ; LAITINEN, Mauri ; WARD, Robert P.: Thinking objectively: software engineering in the small. In: *Commun. ACM* 43 (2000), März, Nr. 3, S. 115–118
- [Fow99] FOWLER, Martin: *Analyse-Muster – Wiederverwendbare Objektmodelle*. Addison Wesley, 1999
- [FS90] FERSTL, Otto K. ; SINZ, Elmar J.: Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). In: *Wirtschaftsinformatik* 32 (1990), Nr. 6, S. 566–581
- [FS91] FERSTL, Otto K. ; SINZ, Elmar J.: Ein Vorgehensmodell zur Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM). In: *Wirtschaftsinformatik* 33 (1991), Nr. 6, S. 477–491
- [FS95] FERSTL, Otto K. ; SINZ, Elmar J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen. In: *Wirtschaftsinformatik* 37 (1995), Nr. 3, S. 209–220
- [FS06] FERSTL, Otto K. ; SINZ, Elmar J.: *Grundlagen der Wirtschaftsinformatik*. Bd. 1. Oldenbourg Verlag, 2006
- [FV99] FERRÉ, X. ; VEGAS, S.: An evaluation of domain analysis methods. (1999)
- [FWLB10] FERNANDEZ, D. M. ; WAGNER, S. ; LOCHMANN, K. ; BAUMANN, A.: Field Study on Requirements Engineering Artefacts and Patterns. In: *Evaluation and Assessment in Software Engineering*, 2010
- [Gar02] GARSHOL, Lars M.: What are Topic Maps? In: *XML.com* (2002)
- [Gar04] GARSHOL, Lars M.: Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. In: *Journal of Information Science* 30 (2004), Nr. 4, S. 378–391
- [GG95] GUARINO, N. ; GIARETTA, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing* (1995), S. 25–32
- [GK95] GOMAA, H. ; KERSCHBERG, L.: Domain modeling for software reuse and evolution. In: *Computer-Aided Software Engineering, 1995. Proceedings., Seventh International Workshop on*, 1995, S. 162–171
- [Glo11] GLOGER, Boris: *Scrum: Produkte zuverlässig und schnell entwickeln*. Hanser Verlag, 2011
- [Góm12] GÓMEZ, Jorge Carlos M.: Serviceorientierte Architektur. In: *Ezyklopädie der Wirtschaftsinformatik – Online Lexikon* (2012)
- [GO94] GRUBER, Thomas R. ; OLSEN, Gregory R.: An Ontology for Engineering Mathematics. In: *KR-94*, 1994, S. 258–269
- [GOS09] *Kapitel What Is an Ontology?* In: GUARINO, Nicola ; OBERLE, Daniel ; STAAB, Steffen: *Handbook of Ontologies*. Springer, 2009, S. 1–18
- [GP99] GÓMEZ-PÉREZ, Asunción: Evaluation of Taxonomic Knowledge in

- Ontologies and Knowledge Bases. In: *Banff Knowledge Acquisition for Knowledge-Based Systems, KAW'99* Bd. 2. Banff, Alberta, Canada : University of Calgary, Alberta, Canada, 1999, 6.1.1–6.1.18
- [GPFV96] GÓMEZ-PÉREZ, A. ; FERNÁNDEZ, M. ; VICENTE, A. de: Towards a Method to Conceptualize Domain Ontologies. In: *12th European Conference on Artificial Intelligence*, 1996
- [GPJP95] GÓMEZ-PÉREZ, Asunción ; JURISTO, Natalia ; PAZOS, Juan: Evaluation and Assessment of the Knowledge Sharing Technology. In: *Towards very large knowledge bases*. Holanda : IOS Press, 1995. – ISBN 90–51992173, S. 289–296
- [Gru93] GRUBER, Thomas R.: A translation approach to portable ontology specifications. In: *Knowledge Acquisition* 5 (1993), S. 199–220
- [Gru95a] GRUBER, T. R.: Toward principles for the design of ontologies used for knowledge sharing. In: *Journal of Human-Computer Studies* 43 (1995), S. 907–928
- [Gru95b] GRUBER, Thomas R.: Toward principles for the design of ontologies used for knowledge sharing. In: *Journal of Human-Computer Studies* 43 (1995), S. 907–928
- [Gua97] GUARINO, Nicola: Semantic matching: Formal ontological distinctions for information organization, extraction, and integration. In: PAZIENZA, MariaTeresa (Hrsg.): *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology* Bd. 1299. Springer Berlin Heidelberg, 1997. – ISBN 978–3–540–63438–6, S. 139–170
- [Gua98] GUARINO, Nicola: Formal Ontology and Information Systems, IOS Press, 1998, S. 3–15
- [HA95] HJØRLAND, Birger ; ALBRECHTSEN, Hanne: Toward a new horizon in information science: Domain-analysis. In: *Journal of the American Society for Information Science* 46 (1995), Nr. 6, S. 400–425
- [Has11] HASENKOPF, Andreas: *Kontextabhängige Vorschläge zur domänenspezifischen Terminologiewahl in der Anforderungserfassung*. Technische Universität München, 2011
- [Hef08] HEFKE, Mark: *Ontologiebasierte Werkzeuge zur Unterstützung von Organisationen bei der Einführung und Durchführung von Wissensmanagement*, Fakultät für Wirtschaftswissenschaften der Universität Fridericiana zu Karlsruhe, Diss., 2008
- [HJD11] HULL, Elizabeth ; JACKSON, Ken ; DICK, Jeremy: *Requirements Engineering*. Springer, 2011
- [HM08] HESSE, Wolfgang ; MAYR, Heinrich C.: Modellierung in der Softwaretechnik: eine Bestandsaufnahme. In: *Informatik-Spektrum* 31 (2008), S. 377–393. – ISSN 0170–6012
- [HSS96] HUBER, Franz ; SCHÄTZ, Bernhard ; SCHMIDT, Alexander ; SPIES, Katharina: AutoFocus: A Tool for Distributed Systems Specification. In: *Proceedings of the 4th International Symposium on Formal Techniques in Real-*

- Time and Fault-Tolerant Systems*. London, UK, UK : Springer-Verlag, 1996 (FTRTFT '96). – ISBN 3–540–61648–9, 467–470
- [HSW97] HEIJST, G. van ; SCHREIBER, A. T. ; WIELINGA, B. J.: Using explicit ontologies in KBS development. In: *Int. J. Hum.-Comput. Stud.* 46 (1997), Nr. 2-3, S. 183–292. – ISSN 1071–5819
- [IEE98] IEEE Standards Board: *IEEE Standard 830-1998. IEEE Recommended Practice for Software Requirements Specification*. 1998
- [IWA91] ISCOE, Neil ; WILLIAMS, Gerald B. ; ARANGO, Guillermo: Domain modeling for software engineering. (1991), S. 340–343. ISBN 0–89791–391–4
- [JCA92] JACOBSON, Ivar ; CHRISTERSON, Magnus ; (AUTOR), Patrik J.: *Object-Oriented Software Engineering. A Use Case Driven Approach*. Addison-Wesley, 1992
- [JU99] JASPER, Robert ; USCHOLD, Mike: A framework for understanding and classifying ontology applications. (1999)
- [JZ93] JACKSON, M. ; ZAVE, P.: Domain descriptions. In: *Requirements Engineering, 1993., Proceedings of IEEE International Symposium on*, 1993, S. 56–64
- [KE10] KUSS, Alfred ; EISEND, Martin: *Marktforschung: Grundlagen der Datenerhebung und Datenanalyse*. Gabler, 2010
- [KO05] KOSCHMIDER, Agnes ; OBERWEIS, Andreas: Ontology based Business Process Description. In: *In: Proceedings of the CAiSE'05 WORKSHOPS*, Springer, 2005, S. 321–333
- [KS09] KNEER, Georg ; SCHROER, Markus: *Handbuch Soziologische Theorien*. VS Verlag für Sozialwissenschaften, 2009
- [Lar05] LARMAN, Craig: *UML 2 und Patterns angewendet*. mitp, 2005
- [Leh07] LEHMANN, Frank: *Integrierte Prozessmodellierung mit ARIS*. dpunkt.verlag, 2007
- [Ler10] LERMAN, Julia: *Programming Entity Framework*. O'Reilly, 2010
- [Lid11] *Kapitel Model-Driven Software Development*. In: LIDDLE, Stephen W.: *Handbook of Conceptual Modeling: Theory, Practice, and Research Challenges*. Springer, 2011, S. 17–54
- [LJ00] LU, Ruqian ; JIN, Zhi: *Domain Modeling-based Software Engineering: A Formal Approach*. Kluwer Academic Publishers, 2000
- [LM97] LAM, W. ; MCDERMID, J. A.: A summary of domain analysis experience by way of heuristics. In: *Proceedings of the 1997 symposium on Software reusability*. New York, NY, USA : ACM, 1997 (SSR '97). – ISBN 0–89791–945–9, 54–64
- [LSR07] LINDEN, Frank J. d. ; SCHMID, Klaus ; ROMMES, Eelco: *Software Product Lines in Action – The Best Industrial Practice in Product Line Engineering*. Springer, 2007
- [LTGP04] LOZANO-TELLO, Adolfo ; GÓMEZ-PÉREZ, Asunción: *ONTOMETRIC*:

- A Method To Choose The Appropriate Ontology. In: *Journal of Database Management*. 2004
- [MA99] Kapitel Categorization. In: MEDIN, Douglas L. ; AGUILAR, Cynthia: *The MIT Encyclopedia of Cognitive Sciences*. The MIT Press, 1999, S. 104–106
- [MB05] MILES, Alistair ; BRICKLEY, Dan: *SKOS Core Guide*. W3C, November 2005
- [Mel08] MELZER, Ingo: *Service-orientierte Architekturen mit Web Services: Konzepte – Standards – Praxis*. Spektrum Akademischer Verlag, 2008
- [MFPKB10] MÉNDEZ FERNÁNDEZ, Daniel ; PENZENSTADLER, Birgit ; KUHRMANN, Marco ; BROY, Manfred: A Meta Model for Artefact-Oriented: Fundamentals and Lessons Learned in Requirements Engineering. In: *Model Driven Engineering Languages and Systems*. Springer Berlin Heidelberg, 2010 (Lecture Notes in Computer Science), S. 183–197
- [MFRW00] MCGUINNESS, Deborah L. ; FIKES, Richard ; RICE, James ; WILDER, Steve: An Environment for Merging and Testing Large Ontologies, Morgan Kaufmann, 2000, S. 483–493
- [MGR95] MANDL, Heinz ; GRUBER, Hans ; RENKL, Alexander: *Situiertes Lernen in multimedialen Lernumgebungen*. (1995)
- [Mie01] MIELKE, Carsten: *Geschäftsprozesse: UML-Modellierung und Anwendungs-Generierung*, Ruhr-Universität Bochum, Diss., 2001
- [MNV02] MISSIKOFF, Michele ; NAVIGLI, Roberto ; VELARDI, Paola: *The Usable Ontology: An Environment for Building and Assessing a Domain Ontology*. (2002)
- [Mor10] MORELLI, Frank: *Geschäftsprozessmodellierung ist tot – lang lebe die Geschäftsprozessmodellierung!* / Hochschule Pforzheim. 2010 (135). – Forschungsbericht
- [MS02] MAEDCHE, Er ; STAAB, Steffen: *Measuring Similarity between Ontologies*. (2002)
- [Mus92] MUSEN, Mark A.: Dimensions of knowledge sharing and reuse. In: *Computers and Biomedical Research* 25 (1992), Nr. 5, S. 435 – 467. – ISSN 0010–4809
- [MW98] MCGUINNESS, Deborah L. ; WRIGHT, Jon R.: Conceptual modelling for configuration: A description logic-based approach. In: *Artif. Intell. Eng. Des. Anal. Manuf.* 12 (1998), S. 333–344. – ISSN 0890–0604
- [NE00] NUSEIBEH, Bashar ; EASTERBROOK, Steve: Requirements engineering: a roadmap. In: *Proceedings of the Conference on The Future of Software Engineering*. New York, NY, USA : ACM, 2000 (ICSE '00). – ISBN 1–58113–253–0, S. 35–46
- [NFF+91] NECHES, Robert ; FIKES, Richard ; FININ, Tim ; GRUBER, Tom ; PATIL, Ramesh ; SENATOR, Ted ; SWARTOUT, William R.: Enabling technology for knowledge sharing. In: *AI Mag.* 12 (1991), S. 36–56. – ISSN 0738–4602
- [NL03] NEILL, C.J. ; LAPLANTE, P.A.: *Requirements engineering: the state of the*

- practice. In: *Software, IEEE* 20 (2003), nov.-dec., Nr. 6, S. 40 – 45. – ISSN 0740–7459
- [NM01] NOY, Natalya F. ; MCGUINNESS, Deborah L. ; STANFORD UNIVERSITY (Hrsg.): *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford, CA, 94305: Stanford University, 2001
- [Nor05] NORTH, Klaus: *Wissensorientierte Unternehmensführung: Wertschöpfung durch Wissen*. Gabler, 2005
- [NT95] NONAKA, Ikujiro ; TAKEUCHI, Irotaka: *The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation*. Oxford University Press, 1995
- [NTK00] NONAKA, Ikujiro ; TOYAMA, Ryoko ; KONNO, Noboru: SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation. In: *Long Range Planning* 33 (2000), Nr. 1, S. 5 – 34
- [NTM12] NONAKA, Ikujiro ; TAKEUCHI, Hirotaka ; MADER, Friedrich: *Die Organisation des Wissens: Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen*. Campus Verlag, 2012
- [OAS06] OASIS OPEN ; OASIS OPEN (Hrsg.): *Reference Model for Service Oriented Architecture 1.0*. Ad.: OASIS Open, August 2006
- [Obj06] OBJECT MANAGEMENT GROUP: *Business Process Model and Notation Specification*. Februar 2006
- [OF06] OSTERLOH, Margit ; FROST, Jetta: *Prozessmanagement als Kernkompetenz: Wie Sie Business Reengineering strategisch nutzen können*. Gabler Verlag, 2006
- [Off02] OFFEN, Ray: Domain Understanding is the Key to Successful System Development. In: *Requirements Engineering* 7 (2002), S. 172–175. – ISSN 0947–3602
- [OR89] OGDEN, C. K. ; RICHARDS, I. A.: *The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism*. HBJ, 1989
- [Par10] PARTSCH, Helmuth: *Requirements-Engineering systematisch*. Springer, 2010
- [PD88] PRIETO-DIAZ, R.: Software reuse: emerging technology. Los Alamitos, CA, USA : IEEE Computer Society Press, 1988. – ISBN 0–8186–0846–3, Kapitel Domain analysis for reusability, S. 347–353
- [PD90] PRIETO-DÍAZ, Rubén: Domain analysis: an introduction. In: *SIGSOFT Softw. Eng. Notes* 15 (1990), Nr. 2, S. 47–54. – ISSN 0163–5948
- [PDA91] PRIETO-DIAZ, Ruben ; ARANGO, G.: *Domain Analysis and Software Systems Modeling*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1991. – ISBN 081868996X
- [PLS73] PEIGH, Pamela S. ; LEE, Thomas H. ; STACHOWIAK, Herbert: *Allgemeine Modelltheorie*. Springer Vienna, 1973
- [PM04] PORZEL, Robert ; MALAKA, Rainer: A Task-based Approach for Onto-

- logy Evaluation. In: *ECAI 2004 Workshop Ontological Learning and Population*, 2004
- [PN02] PEASE, Adam ; NILES, Ian: IEEE standard upper ontology: a progress report. In: *The Knowledge Engineering Review* 17 (2002), Nr. 01, S. 65–70
- [Pol66] POLANYI, Michael: *The Tacit Dimension*. Doubleday & Company, Inc., 1966
- [PRR06] PROBST, Gilbert ; RAUB, Steffen ; ROMHARDT, Kai: *Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. Gabler, 2006
- [Rat09] RATIU, Daniel: *Intentional Meaning of Programs*, Technische Universität München, Diss., 2009
- [RBP+91] RUMBAUGH, James ; BLAHA, Michael ; PREMERLANI, William ; EDDY, Frederick ; LORENSON, William: *Object-Oriented Modeling and Design*. Prentice Hall, 1991
- [RF10] RÜCKER, Bernd ; FREUND, Jakob: *Praxishandbuch BPMN 2.0*. Hanser, 2010
- [RGE+94] ROTHENFLUH, Thomas E. ; GENNARI, John H. ; ERIKSSON, Henrik ; PUERTA, Angel R. ; TU, Samson W. ; MUSEN, Mark A.: Reusable Ontologies, Knowledge-Acquisition Tools, and Performance Systems: PROTÉGÉ-II Solutions to Sisyphus-2. In: *International Journal of Human-Computer Studies*, 1994, S. 43–1
- [RH09] RUNESON, Per ; HÖST, Martin: Guidelines for conducting and reporting case study research in software engineering. In: *Empirical Software Engineering* 14 (2009), S. 131–164. – ISSN 1382–3256
- [RK96] REHÄUSER, Jakob ; KRCCMAR, Helmut: *Wissensmanagement im Unternehmen / Universität Hohenheim, Lehrstuhl für Wirtschaftsinformatik*. 1996. – Forschungsbericht
- [RS09] RUPP, Chris ; SOPHIST GROUP: *Requirements-Engineering und -Management*. Hanser, 2009
- [Rät98] RÄTZMANN, Manfred: *Three-Tier-Development*. http://www.dfpug.de/konf/konf_1998/09_tier/d_tier/d_tier.htm, Besucht am 2013-06-02, 1998
- [Rum11] RUMPE, Bernhard: *Modellierung mit UML: Sprache, Konzepte und Methodik*. Springer, 2011
- [SBF98] STUDER, Rudi ; BENJAMINS, V.Richard ; FENSEL, Dieter: Knowledge engineering: Principles and methods. In: *Data and Knowledge Engineering* 25 (1998), S. 161 – 197. – ISSN 0169–023X
- [sBF+10] ÖSTERLE, Hubert ; BECKER, Jörg ; FRANK, Ulrich ; HESS, Thomas ; KARAGIANNIS, Dimitris ; KRCCMAR, Helmut ; LOOS, Peter ; MERTENS, Peter ; OBERWEIS, Andreas ; SINZ, Elmar J.: Memorandum zur gestaltungsorientierten Wirtschaftsinformatik. In: *Zeitschrift für betriebswirtschaftliche Forschung* 6 (2010), S. 664–672
- [Sch00] SCHMID, Klaus: Scoping software product lines: an analysis of an emer-

- ging technology. In: *Proceedings of the first conference on Software product lines : experience and research directions: experience and research directions*. Norwell, MA, USA : Kluwer Academic Publishers, 2000. – ISBN 0-79237-940-3, 513-532
- [Sch02] SCHEER, August-Wilhelm: *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. Springer, 2002
- [Sch12] SCHEWE, Gerhard: *Gablers Wirtschaftslexikon: Organisationsmodell*. <http://wirtschaftslexikon.gabler.de/Definition/organisationsmodell.html>, 2012
- [Sel12] SELIC, Bran: What will it take? A view on adoption of model-based methods in practice. In: *Software and Systems Modeling* 11 (2012), S. 513-526. – ISSN 1619-1366
- [Sin97] SINZ, Elmar J.: *Architektur betrieblicher Informationssysteme*. Otto-Friedrich-Univ., 1997
- [SN00] SCHEER, August-Wilhelm ; NÜTTGENS, Markus: ARIS Architecture and Reference Models for Business Process Management. In: AALST, Wil (Hrsg.) ; DESEL, Jörg (Hrsg.) ; OBERWEIS, Andreas (Hrsg.): *Business Process Management* Bd. 1806. Springer Berlin Heidelberg, 2000. – ISBN 978-3-540-67454-2, S. 376-389
- [SS09] STAAB, Steffen ; STRUDER, Rudi: *Handbook on Ontologies*. Springer, 2009
- [Sta73] STACHOWIAK, H.: *Allgemeine Modelltheorie*. Springer-Verlag, 1973. – ISBN 9783211811061
- [Ste71] STEVER, Hermann: *Superierung durch Komplexbildung*, Universität Karlsruhe, Diss., 1971
- [Ste93] STEINMÜLLER, Wilhelm: *Informationstechnologie und Gesellschaft: Einführung in die angewandte Informatik*. Wissenschaftliche Buchgesellschaft Darmstadt, 1993
- [Str90] *Kapitel Information, Wissen und Bedeutung: Eine Analyse systemischer Strukturen sprachlicher Kommunikation*. In: STROHNER, Hans: *Information ohne Kommunikation*. Fischer Taschenbuch Verlag, 1990, S. 209-226
- [sWB10] ÖSTERLE, Hubert ; WINTER, Robert ; BRENNER, Walter: *Gestaltungsorientierte Wirtschaftsinformatik: Ein Plädoyer für Rigor und Relevanz*. University of St. Gallen, 2010
- [Tra94] TRACZ, Will: Domain-specific software architecture (DSSA) frequently asked questions (FAQ). In: *SIGSOFT Softw. Eng. Notes* 19 (1994), Nr. 2, S. 52-56. – ISSN 0163-5948
- [TT28] THOMAS, W.I. ; THOMAS, D.S.T.: *The child in America: behavior problems and programs*. Johnson Reprint Corp., 1928
- [UG96] USCHOLD, Mike ; GRUNINGER, Michael: Ontologies: principles, methods and applications. In: *The Knowledge Engineering Review* 11 (1996), Nr. 02, S. 93-136

- [UKMZ98] USCHOLD, Mike ; KING, Martin ; MORALEE, Stuart ; ZORGIOS, Yannis: The Enterprise Ontology. In: *The Knowledge Engineering Review* 13 (1998), 2, S. 31–89. – ISSN 1469–8005
- [Ull79] ULLMANN, Stephen: *Semantics: An Introduction to the Science of Meaning*. Rowman & Littlefield, 1979
- [Ven99] VENNIX, J. A. M.: Group Model Building : tackling messy problems. In: *System Dynamics Review* 15 (1999), S. 379–401
- [Wal38] WALDE, Alois: *Lateinisches etymologisches Wörterbuch*. University of Michigan, 1938
- [Wil00] WILLIAMS, Laurie A.: *The Collaborative Software process*, University of Utah – Department of Computer Science, Diss., 2000
- [WLS12] WOHLTMANN, Hans-Werner ; LACKES, Richard ; SIEPERMANN, Markus: *Daten*. Gablers Wirtschaftslexikon, 2012
- [WRH⁺12] WOHLIN, Claes ; RUNESON, Per ; HÖST, Martin ; OHLSSON, Magnus C. ; REGNELL, Björn ; WESSLÉN, Anders: *Experimentation in Software Engineering*. Springer, 2012
- [WSBK08] WOLTER, Katharina ; ŚMIAŁEK, Michał ; BILDHAUER, Daniel ; KAINDL, Hermann: Reusing Terminology for Requirements Specifications from WordNet. In: *RE '08: Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*. Washington, DC, USA : IEEE Computer Society, 2008. – ISBN 978–0–7695–3309–4, S. 325–326
- [WW07] WILKESMANN, Uwe ; WÜRMSEER, Grit: *Wissensmanagement an Universitäten / Universität Dortmund*. 2007. – Forschungsbericht
- [Zal12] ZALTA, Edward N.: *Stanford Encyclopedia of Philosophy*. Center of the Study of Language and Information, Stanford University, 2012 <http://plato.stanford.edu/>
- [ZV07] ZHOA, Liping ; VERSCHUEREN, Paul: Domänenanalyse mit wiederverwendbaren Problemmustern. In: *bit-fabrik* (2007), S. 70–77