

Ein Modellierungsansatz für die Umgestaltung von Anwendungslandschaften

Dissertation zur Erlangung des Doktorgrades
an der Fakultät für Mathematik, Informatik und Naturwissenschaften
Fachbereich Informatik
der Universität Hamburg

vorgelegt von
Stefan Hofer

Hamburg, 2017

Tag der Disputation: 24. April 2017

Folgende Gutachter empfehlen die Annahme der Dissertation:

Prof. Dr.-Ing. Heinz Züllighoven

o. Univ.-Prof. DI Dr. Gustav Pomberger

Prof. Dr.-Ing. Matthias Riebisch

Abstract

To support their business processes, companies run *application landscapes*. These consist of software systems that are in part dependent on each other. Application landscapes need to be transformed frequently to adapt to a changing technological, legal, and economical environment. Often, changes to an application landscape's software systems affect business processes. Such dependencies between an application landscape's architecture and its use have to be transparent to transform an application landscape successfully.

Models are commonly used to facilitate the transformation of application landscapes. However, existing modeling approaches fall short of this purpose. Especially, technological and domain-focused views of application landscapes are often poorly integrated.

This thesis provides an empirical analysis of modeling-related problems in transformation projects. The author identifies which requirements should be met by a modeling approach for application landscape transformation. These requirements target modeling language, modeling methodology and how models should be used. Existing modeling approaches are evaluated against the requirements to assess their strengths and weaknesses. These weaknesses are addressed in the constructive part of this thesis. Tried and proven techniques and language features of existing approaches are combined so that the requirements are met. To achieve re-usability, the language features and modeling techniques are generalized. The resulting „building blocks“ can be used to adapt existing approaches to transformation projects

A proof-of-concept demonstrates how to apply the building blocks to a commercial modeling approach. An evaluation of the building blocks shows that they are applicable and relevant to the transformation of application landscapes.

Zusammenfassung

Um ihre Geschäftsprozesse zu unterstützen, betreiben Unternehmen *Anwendungslandschaften*, die aus teilweise voneinander abhängigen Softwaresystemen bestehen. Anwendungslandschaften müssen immer wieder tiefgreifend verändert werden, um sie an technische, rechtliche und wirtschaftliche Randbedingungen anzupassen. Oft wirken sich Änderungen an Softwaresystemen auf die Geschäftsprozesse aus. Damit eine Anwendungslandschaft erfolgreich umgestaltet werden kann, müssen die Zusammenhänge zwischen ihrer Architektur und ihrem Einsatz verstanden werden.

In der Praxis werden Modelle als Hilfsmittel für die Umgestaltung von Anwendungslandschaften eingesetzt. Existierende Modellierungsansätze eignen sich jedoch nur eingeschränkt für diesen Zweck, da sie technologische und fachliche Sichten auf Anwendungslandschaften nicht hinreichend verzahnen.

In dieser Arbeit werden die Probleme mit der Modellierung in Umgestaltungsprojekten empirisch untersucht und Anforderungen an Modellierungsansätze abgeleitet. Die Anforderungen beziehen sich auf die Modellierungssprache, auf das Modellierungsvorgehen und auf den Einsatz von Modellen. Ausgewählte Modellierungsansätze werden bewertet und so ihre Stärken und Schwächen aufgezeigt. Im konstruktiven Teil der Arbeit werden die Stärken der untersuchten Ansätze miteinander kombiniert, sodass die Anforderungen erfüllt werden. Um dieses Ziel zu erreichen, werden Sprachmittel und Vorgehensweisen, die sich gut für Umgestaltungsprojekte eignen, zu wiederverwendbaren „Bausteinen“ verallgemeinert.

Um einen existierenden Ansatz an Umgestaltungsprojekte anzupassen, muss dieser gegen die Anforderungen abgeglichen und die identifizierte Lücke durch geeignete Bausteine geschlossen werden. Dies wird beispielhaft gezeigt, indem mehrere Bausteine in einen kommerziellen Modellierungsansatz integriert werden. Eine Evaluation der Bausteine, die neben anderen Indikatoren auf einem Praxisbeispiel beruht, zeigt ihre Relevanz, Anwendbarkeit und Praxistauglichkeit.

Danksagungen

Viele Menschen haben mich dabei unterstützt, diese Arbeit zu schreiben. Bei ihnen möchte ich mich herzlich bedanken.

Ich danke den Professoren, die meine Arbeit betreuten und begutachteten: Heinz Züllighoven verstand es, meine vagen Ideen zu verdichten. Gustav Pomberger vermittelte mir, wie man eine methodisch und stilistisch saubere Arbeit zu Papier bringt. Matthias Riebisch unterstützte mich „auf den letzten Metern“.

Der Geschäftsleitung der WPS – Workplace Solutions GmbH danke ich dafür, dass sie mich in Projekten eingesetzt hat, die für diese Arbeit relevant waren. Mit eurer unkomplizierten Art habt ihr mir Vieles erleichtert.

Den Mitarbeiterinnen und Mitarbeitern des – mittlerweile aufgelösten – Arbeitsbereichs Softwaretechnik der Universität Hamburg danke ich dafür, dass sie mir einen „Heimathafen“ geboten haben. Auf unseren Klausurtagen konnte ich ungemein von euch profitieren. Besonders bedanke ich mich bei meinen "Leidensgenossen". Ihr gabt mir das gute Gefühl, nicht alleine zu sein.

Die gemeinsamen Tage mit Dr. Eugen Reisch in der Staatsbibliothek Hamburg gehörten zu meinen produktivsten. Auch nach unserer gemeinsamen Zeit im Lesesaal blieb Eugen mir als Wegbegleiter erhalten. Eugen, dass du mit deinen Kommentaren oft Recht hattest, war für mich nicht immer leicht zu akzeptieren. Aber wenn ich nun auf alles zurückblicke, fällt mein Dank an dich umso größer aus.

Carola, Ute, Arne, Johannes und Holger: Ihr standet mir als Interviewpartner zur Verfügung. Viele weitere Menschen haben sich die Zeit genommen, um meine Fragebögen auszufüllen. Euch und Ihnen danke ich für die Bereitschaft, wertvolle Erfahrungen mit mir zu teilen.

Zuletzt danke ich meinen Freunden und meiner Familie für euer Verständnis, dass ich in den letzten Jahren mehr Zeit mit dieser Arbeit verbracht habe als mit euch.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Die untersuchten Fragestellungen	3
1.2	Einordnung der Arbeit	4
1.3	Forschungsmethodik	8
1.3.1	Design Research	8
1.3.2	Ergebnisse	10
1.4	Aufbau der Arbeit	14
2	Umgestaltung von Anwendungslandschaften	15
2.1	Anwendungslandschaften	15
2.2	Umgestaltungsnotwendigkeiten und Ziele	20
2.3	Umgestaltungsprojekte	23
2.4	Zusammenfassung	28
3	Modellierung in Umgestaltungsprojekten	29
3.1	Modelle von Anwendungslandschaften	29
3.2	Modellierungsansätze	33
3.3	Modellierungsprobleme	34
3.4	Analyse der Modellierungsprobleme	37
3.4.1	Modellierung als Realitätskonstruktion	37
3.4.2	Modelle als Gestaltungsgegenstand	39
3.4.3	Modellierung als Lern- und Kommunikationsprozess	41
3.4.4	Architektursichten	42
3.5	Anforderungen an Modellierungsansätze	45
3.5.1	Anforderungen an die Modellierungssprache	48
3.5.2	Anforderungen an das Modellierungsvorgehen	50

3.5.3	Anforderungen an den Umgang mit Modellen	50
3.6	Zusammenfassung	51
4	Bewertung existierender Modellierungsansätze	53
4.1	Bewertungsziele und Auswahl der Ansätze	53
4.2	Bewertung	56
4.2.1	ArchiMate	56
4.2.2	ARIS	59
4.2.3	BEN	62
4.2.4	EAM Patterns	64
4.2.5	eGPM	65
4.3	Zusammenfassung	67
5	Ein Modellierungsansatz für Umgestaltungsprojekte	69
5.1	Zielvorstellung: Ein idealer Modellierungsansatz	69
5.2	Bausteine eines Modellierungsansatzes	70
5.3	Anwendungsbeispiel	78
5.3.1	Auswahl des zu adaptierenden Modellierungsansatzes	78
5.3.2	Überblick über den Modellierungsansatz	80
5.3.3	Anwendung der Bausteine	84
5.4	Zusammenfassung	96
6	Evaluation der Bausteine	97
6.1	Auswahl von Qualitätsmerkmalen	97
6.2	Bewertung der Qualitätsmerkmale	99
6.2.1	Anwendbarkeit	99
6.2.2	Relevanz	100
6.2.3	Praxistauglichkeit	104
6.2.4	Fazit	107
6.3	Zusammenfassung	109
7	Zusammenfassung und Ausblick	111
A	Empirische Untersuchung von Praxisprojekten	119
A.1	Experteninterviews	119
A.1.1	Einleitung	120

A.1.2	Umgestaltungsszenario <i>Testfälle</i>	121
A.1.3	Umgestaltungsszenario <i>Handlungsbedarf</i>	123
A.1.4	Umgestaltungsszenario <i>Sichten</i>	125
A.1.5	Umgestaltungsszenario <i>Fahrplan</i>	127
A.1.6	Umgestaltungsszenario <i>Planungsperioden</i>	129
A.1.7	Umgestaltungsszenario <i>Migrationsarchitektur</i>	129
A.1.8	Umgestaltungsszenario <i>Bewertung einer Anwendungsland-</i> <i>schaft</i>	132
A.2	Fragebogen	135
B	Bewertung ausgewählter Modellierungsansätze	143
B.1	ArchiMate	145
B.2	ARIS	150
B.3	BEN	156
B.4	EAM Patterns	160
B.5	eGPM	165
C	eGPM-Modelltyp Kooperationsbild	171
D	Anwendungsbeispiel: Erweiterung von ADOit	177
D.1	Kooperationsszenario	178
D.2	Anwendungsarchitektur	183
D.3	Datenarchitektur	184
D.4	Technologiearchitektur	184
E	Evaluation des erweiterten ADOit	189

Abbildungsverzeichnis

1.1	Modellierungsansätze helfen, Anwendungslandschaften und Modelle zu gestalten	4
1.2	Perspektiven auf Anwendungslandschaften (in Grau: Relevanz für diese Arbeit)	5
1.3	Artefakte und Schritte des Design Research	10
1.4	Vorgehen bei der Ausarbeitung der Ergebnisse	12
2.1	Begriffe und ihr Zusammenhang (Teil 1)	17
2.2	Begriffe und ihr Zusammenhang (Teil 2)	18
2.3	Begriffe und ihr Zusammenhang (Teil 3)	18
2.4	Begriffe und ihr Zusammenhang (Teil 4)	19
2.5	Anwendungsorientierte Perspektive auf Anwendungslandschaften .	21
3.1	Umgestaltung einer Anwendungslandschaft mit Hilfe von Modellen (angelehnt an [LL13, S.8])	31
3.2	Abteilungsübergreifender und anwendungsübergreifender Prozess .	36
3.3	Modelle als Sichten auf ein Original	43
4.1	Ausschnitt eines Geschäftsprozessmodells mit ArchiMate (vollständige Grafik siehe Abbildung B.2)	58
4.2	Ausschnitt eines fachlichen Architekturmodells mit ArchiMate (vollständige Grafik siehe Abbildung B.1)	59
4.3	<i>ARIS-Haus</i> , angelehnt an [Sch02, S.41]	60
4.4	Ausschnitt eines Geschäftsprozesses, dargestellt als <i>EPK</i> (vollständige Grafik siehe Abbildung B.6)	62
4.5	Ausschnitt einer fachlichen Architektur einer Anwendungslandschaft, dargestellt mit <i>ADOben</i> (vollständige Grafik siehe Abbildung B.13)	63

4.6	Ausschnitt eines Geschäftsprozessmodells als eGPM Kooperations- bild (vollständige Grafik siehe Abbildung B.23)	67
5.1	Modelltypen eines Modellierungsansatzes für Umgestaltungsprojekte	75
5.2	Das Modellelement „Makler“ in einem Informationsmodell und zwei grafischen Modellen	79
5.3	Beispiel für den Modelltyp Kooperationsszenario	81
5.4	Beispiel für den Modelltyp Datenarchitektur	82
5.5	Beispiel für den Modelltyp Anwendungsarchitektur	83
5.6	Beispiel für den Modelltyp Technologiearchitektur	84
5.7	Arbeitsgegenstand ohne (links) und mit (rechts) offenen Fragen . . .	90
5.8	Vage Information als „offene Frage“ dokumentieren	91
5.9	Modellelemente über Referenzen miteinander verknüpfen	92
5.10	Anwendungsarchitektur mit Zeitfilter	93
5.11	Modellvergleich zwischen einem Ist- und einem Soll-Kooperations- szenario	93
5.12	Sequenz von Tätigkeiten (Ausschnitt aus einem Kooperationsszenario)	94
A.1	Szenario „Testfälle“	122
A.2	Szenario „Handlungsbedarf“	123
A.3	Szenario „Sichten“	126
A.4	Szenario „Fahrplan“	128
A.5	Szenario „Planungsperioden“	130
A.6	Szenario „Migrationsarchitektur“	131
A.7	Szenario „Bewertung einer Anwendungslandschaft“	134
B.1	Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als Kombination der ArchiMate Viewpoints <i>Business Process, Application Usage</i> und <i>Information Infrastructure</i>	147
B.2	Fachliche Architektur (Ist) durch Kombination der ArchiMate View- points <i>Application Cooperation</i> und <i>Application Structure</i>	148
B.3	Geschäftsprozess „Leistungsfall bearbeiten“ (Soll) als Kombination der ArchiMate Viewpoints <i>Business Process, Application Usage</i> und <i>Information Infrastructure</i>	148
B.4	Fachliche Architektur (Soll) durch Kombination der ArchiMate Viewpoints <i>Application Cooperation</i> und <i>Application Structure</i> . . .	149

B.5	Legende aller verwendeten ArchiMate Modellelementtypen und Relationen	149
B.6	Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als ARIS EPK . . .	152
B.7	Fachliche Architektur (Ist) als ARIS Systemlandschaft	153
B.8	Geschäftsprozess „Leistungsfall bearbeiten“ (Soll) als ARIS EPK . . .	154
B.9	Fachliche Architektur (Soll) als ARIS Systemlandschaft	155
B.10	Legende zur ARIS EPK	155
B.11	Legende zur ARIS Systemlandschaft	155
B.12	Fachliche Architektur (Ist) als ADOben Applikationslandkarte . . .	158
B.13	Fachliche Architektur (Soll) als ADOben Applikationslandkarte . . .	158
B.14	Legende zur ADOben Applikationslandkarte	158
B.15	Modell vom Typ ADOben <i>Applikationen (Bestandsführung)</i> – wird von den Modellen B.12 und B.13 referenziert	159
B.16	Modell vom Typ ADOben <i>Datenmodell</i> – wird vom Modell B.15 referenziert	159
B.17	Legende zum EAM-Pattern V-48	162
B.18	Legende zu den EAM-Patterns V-18 und V-51	162
B.19	Fachliche Architektur (Ist) als EAM-Pattern V-48	162
B.20	Fachliche Architektur (Soll) als EAM-Pattern V-48	163
B.21	Ausschnitt aus Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als Kombination der EAM-Pattern V-18 und V-51	163
B.22	Ausschnitt aus Geschäftsprozess „Leistungsfall bearbeiten“ (Soll) als Kombination der EAM-Pattern V-18 und V-51	164
B.23	Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als eGPM Kooperationsbild	167
B.24	Fachliche Architektur (Ist) als eGPM IT-Landschaft	168
B.25	Legende zum Modelltyp Kooperationsbild	168
B.26	Legende zum Modelltyp IT-Landschaft	168
B.27	Geschäftsprozess „Leistungsfall bearbeiten“ (soll) als eGPM Kooperationsbild	169
B.28	Fachliche Architektur (Soll) als eGPM IT-Landschaft	170
B.29	Begriffsmodell (Ist und Soll) als eGPM Begriffsmodell	170
B.30	Legende zum Modelltyp Begriffsmodell	170
C.1	Beispiel für ein Kooperationsbild	174

D.1	Modelltyp Kooperationsszenario	179
D.2	Die Relation „bearbeitet“ als UML Klassendiagramm	180
D.3	Die Relation „initiiert“ als UML Klassendiagramm	180
D.4	Die Relation „kommuniziert über“ als UML Klassendiagramm	180
D.5	Modelltyp Anwendungsarchitektur	185
D.6	Modelltyp Datenarchitektur	186
D.7	Modelltyp Technologiearchitektur	187
E.1	Antragsprozess für eine Krankenversicherung	190
E.2	Regelprüfung im Antragsprozess	191
E.3	Prozess für eine Ehegattenermäßigung	192
E.4	Krankenleistungsprozess	193
E.5	Fachliche Konzepte als Datenarchitektur-Modell	194
E.6	Anwendungslandschaft (KV-Anträge und Leistung) als Anwendungs- architektur-Modell	194
E.7	Technische Referenzarchitektur mit WfMS als Technologiearchitektur- Modell	195

Kapitel 1

Einleitung

Große Unternehmen betreiben heute in der Regel viele, teilweise voneinander abhängige Softwaresysteme – sogenannte *Anwendungslandschaften*. Um mit wechselnden technischen, rechtlichen und wirtschaftlichen Rahmenbedingungen Schritt zu halten, müssen Anwendungslandschaften oft tiefgreifend verändert werden. Solche Veränderungen stellen Unternehmen vor große Schwierigkeiten, wie folgende Beispiele verdeutlichen:

Nach einem Strategiewechsel löst ein Finanzdienstleister seine selbstentwickelten Softwaresysteme durch zugekaufte Standardsoftware ab. Die bestehenden Daten müssen in die Standardsoftware migriert und die Arbeitsabläufe der Mitarbeiter an die Möglichkeiten des neuen Softwaresystems angepasst werden.

Ein Logistikunternehmen will seine Kapazität durch den Zukauf neuer Anlagen erhöhen und muss dazu deren Leitstands-Software in die vorhandene Anwendungslandschaft integrieren. Um tatsächlich eine Kapazitätssteigerung zu erreichen, sind auch Anpassungen im Arbeitsablauf der Mitarbeiter nötig. Eine störungsfreie Inbetriebnahme der Anlagen setzt daher voraus, dass die Leitstands-Software und die Arbeitsabläufe genau aufeinander abgestimmt werden.

Projekte zur Umgestaltung von Anwendungslandschaften sind in der Regel mit großem Aufwand und festen Rahmenbedingungen verbunden – etwa Terminen, bis zu denen gesetzliche Vorgaben umgesetzt sein müssen. Das Scheitern eines solchen Projekts könnte geschäftsschädigende und sogar existenzbedrohende Folgen für

ein Unternehmen haben (vgl. [LSV00, S.118f]). Um eine Umgestaltung zu planen, müssen Informationen über die betroffene Anwendungslandschaft gesammelt und analysiert werden. Die daran beteiligten Personen wie Anwender und IT-Experten nehmen auf Grund ihrer Rollen unterschiedlichen Sichtweisen auf eine Anwendungslandschaft ein. Das kann zu Missverständnissen und Widersprüchen führen. Außerdem besteht das Risiko, dass relevante Aspekte der Anwendungslandschaft unzureichend bei der Umgestaltung berücksichtigt werden.

Um die komplexen Zusammenhänge in einer Anwendungslandschaft zu verstehen und die Auswirkungen einer Umgestaltung zu analysieren und zu kommunizieren, werden in der Praxis *Modelle* eingesetzt:

„Wann immer Menschen etwas Größeres verstehen, schaffen oder unternehmen wollen, so machen sie sich ein (meist vereinfachtes) Bild davon, nehmen Maß, fertigen einen Plan oder eine Skizze an, orientieren sich an einem Vorbild oder bauen sich ein solches — sei es in Gedanken, mit Schreibzeug, Papier, anderem Material oder mit dem Computer. [. . .] In allen Fällen stützen sie sich dabei auf etwas, was man gemeinhin als Modell bezeichnet. Ein Modell ist im weitesten Sinne ein Stellvertreter — für das, was verstanden, geschaffen, unternommen oder betrieben werden soll.“ [HM08, S.377]

In dieser Arbeit wird gezeigt, dass die mit existierenden Ansätzen erstellten Modelle von Anwendungslandschaften die genannten Probleme in Umgestaltungsprojekten nicht ausreichend lösen. Der Bedarf an besser geeigneten Modellierungsansätzen nimmt jedoch tendenziell zu, weil „die [Anwendungs]-Landschaften in und zwischen Unternehmen stetig größer und umfassender werden“ [LZ14]. Das heißt, dass in Unternehmen immer mehr Softwaresysteme nur dann verändert oder in Betrieb genommen werden können, wenn eine existierende Anwendungslandschaft einbezogen wird. Daher müssen den Modellierungsproblemen in Umgestaltungsprojekten mehr Gewicht beigemessen und besser angepasste Modellierungsansätze entwickelt werden. Mit solchen spezialisierten Ansätzen können Unternehmen aussagekräftigere Modelle ihrer Anwendungslandschaft erstellen und damit Umgestaltungsprojekte besser unterstützen. Ziel dieser Arbeit ist nicht, den *einen* Modellierungsansatz für Umgestaltungsprojekte zu entwickeln. Stattdessen sollen die für Umgestaltungsprojekte Verantwortlichen in die Lage versetzt werden, existierende Ansätze besser an ihre Umgestaltungsprojekte anzupassen.

1.1 Die untersuchten Fragestellungen

Diese Arbeit leistet einen Beitrag zur Entwicklung von Modellierungsansätzen für Umgestaltungsprojekte. Dazu muss die Umgestaltung von Anwendungslandschaft zunächst als Diskursbereich etabliert werden. *Umgestaltungsprojekt* wird in dieser Arbeit als ein neuer Typ von Projekt eingeführt und anhand folgender Forschungsfragen untersucht:

- Was charakterisiert Umgestaltungsprojekte?
- Was wird in solchen Projekten mit Modellen bezweckt?
- Welche Probleme gibt es in solchen Projekten bei der Modellierung?

Diese Fragen wurden mit Hilfe einer empirischen Untersuchung und anhand von Literatur beantwortet. Damit das gesteckte Ziel erreicht werden kann, bedarf es nicht nur eines analytischen, sondern auch eines konstruktiven Beitrags zur Modellierung in Umgestaltungsprojekten. Dazu müssen die Probleme der Modellierung und ihre möglichen Lösungen genauer betrachtet werden. Daher stellt sich die Frage:

- Welche Gründe gibt es für die festgestellten Modellierungsprobleme?

In dieser Arbeit werden die Modellierungsprobleme auf konzeptuellen Grundlagen der Modellierung in Umgestaltungsprojekten zurückgeführt und eine neue Herangehensweisen an die Modellierung in Umgestaltungsprojekten vorgeschlagen. Dazu gehört unter anderem die Feststellung, dass Anwendungslandschaften und ihr Einsatz in menschlichen Arbeitsabläufen nicht vollständig erfasst und in Modellen abgebildet werden können. Solche Modelle zeigen keine objektiven Fakten. Vielmehr muss Modellierung als ein sozialer Prozess verstanden werden, in dem die Beteiligten unterschiedliche Sichten auf eine Anwendungslandschaft haben und sich mit Modellen ihre eigene „Realität“ konstruieren. Modelle sind daher nicht nur Hilfsmittel für die Umgestaltung, sondern genauso Gestaltungsgegenstand wie die Anwendungslandschaften selbst (siehe Abbildung 1.1).

Die letzte Forschungsfrage befasst sich damit, wie die konzeptionellen Grundlagen der Modellierung dazu beitragen, verbesserte Modellierungsansätze zu entwickeln:

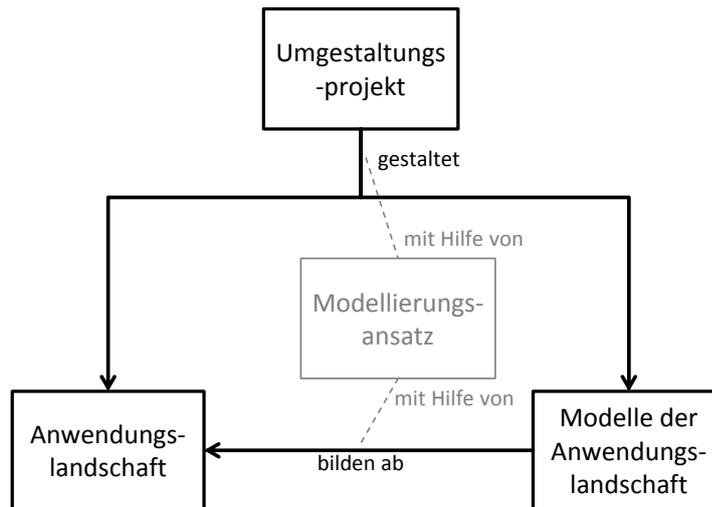


Abbildung 1.1: Modellierungsansätze helfen, Anwendungslandschaften und Modelle zu gestalten

- Welche Mittel muss ein Modellierungsansatz mitbringen, damit er sich besonders für Umgestaltungsprojekte eignet?

Diese Frage wird beantwortet, indem aus den konzeptionellen Grundlagen konkrete Anforderungen an Modellierungsansätze abgeleitet werden. Anhand dieser Anforderungen werden existierende Modellierungsansätze bewertet und dabei Konzepte identifiziert, welche für Umgestaltungsprojekte besonders nützlich sind. Diese Konzepte werden zu „Bausteinen“ verdichtet, mit denen existierende Modellierungsansätze ergänzt werden können, sodass sie mehr Anforderungen erfüllen.

1.2 Einordnung der Arbeit

Anwendungslandschaften werden in der Literatur häufig aus einer betriebswirtschaftlichen (vgl. z.B. [Han13] und [Win10]) und einer technologischen Perspektive (vgl. z.B. [EHH⁺08]) betrachtet. Diese Arbeit führt zusätzlich eine anwendungsorientierte Perspektive ein. Abbildung 1.2 setzt die verschiedenen Perspektiven zu einander in Beziehung und zeigt, wie stark in dieser Arbeit auf sie eingegangen wird.

Die Diskussion der betriebswirtschaftlichen Perspektive wird von der „Ausrichtung der IT an den Geschäftszielen des Unternehmens“ [Kel12, S.36], dem sogenannten *Business-IT-Alignment* dominiert. Von der Unternehmensstrategie über Geschäftsprozesse bis zur Informationstechnologie erfolgt der Blick auf Anwendungslandschaften „top down“. Ein wichtiges Mittel zum Business-IT-Alignment

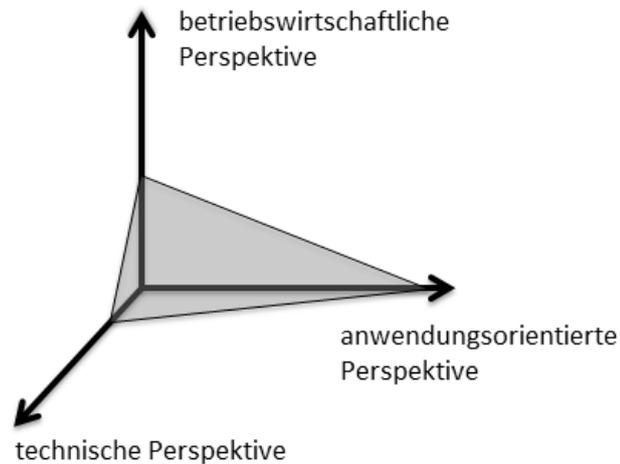


Abbildung 1.2: Perspektiven auf Anwendungslandschaften (in Grau: Relevanz für diese Arbeit)

ist die sogenannte *Unternehmensarchitektur*¹ (vgl. [ARW08]). Sie „befasst sich mit allen Aspekten, die benötigt werden, um die IT-Funktion eines Unternehmens zu beschreiben und zu managen“ [Kel12, S.24]. Die damit verfolgten Ziele umfassen unter anderem die Verbesserung der Kundenzufriedenheit, schnellere Einführung von Produkten und Kostenreduktion (vgl. [Kel12]). Die betriebswirtschaftliche Perspektive auf Anwendungslandschaft schließt auch eine Reihe von ständigen Managementprozessen wie beispielsweise das *IT-Anwendungsportfolio-Management* und das *IT-Projektportfoliomanagement* ein: Auf Basis von regelmäßig zu aktualisierenden Daten über die Anwendungslandschaft werden Entscheidungen über umzusetzende Maßnahmen getroffen, etwa bestimmte Softwaresysteme abzulösen. Daher können die im Rahmen des Business-IT-Alignment durchgeführten Managementprozesse die Umgestaltung von Anwendungslandschaften auslösen.

Die betriebswirtschaftliche Perspektive alleine reicht für diese Arbeit nicht aus, weil Unternehmensarchitekturen nicht als Hilfsmittel für die detaillierte Planung und Durchführung von Änderungen in Anwendungslandschaften gedacht sind. Unternehmensarchitekturen sollen laut Winter et al. (vgl. [ARW08] und [WF06]) auf aggregierten Informationen aufbauen und strategisch ausgerichtet sein. Für operative Aufgaben werden spezialisierte Ansätze empfohlen. Daraus wird klar, dass Unternehmensarchitekturen nicht genügend Detailinformationen für die operative Umgestaltung von Anwendungslandschaften enthalten. Umgestaltung erfordert,

¹Oft auch als *Enterprise Architecture*, *Enterprise IT Architecture* oder *IT-Unternehmensarchitektur* bezeichnet.

Anwendungslandschaften im Kontext ihres Einsatzes in Geschäftsprozessen zu betrachten. Das Bild einer Anwendungslandschaft ergibt sich bei dieser *anwendungsorientierten* Betrachtungsweise „bottom up“ aus den Arbeitsabläufen, die Akteure mit Hilfe von Softwaresystemen verrichten. Daher muss in dieser Arbeit auch eine anwendungsorientierte Perspektive eingenommen werden.

Anwendungsorientierung ist ein Konzept aus der Softwaretechnik und liegt dem *Werkzeug & Material-Ansatz* (vgl. [Zül98] und [Zül05]) zur Software-Entwicklung zu Grunde. Anwendungsorientierung berücksichtigt die Anwender, ihre Aufgaben und die Geschäftsprozesse, an denen sie beteiligt sind (vgl. [Zül05, S.4]).

In der Softwaretechnik beeinflusst die Anwendungsorientierung die Konstruktion von Softwaresystemen, den Entwicklungsprozess und die dafür nötigen Dokumente und Modelle. Sie hat sich bei der Entwicklung von Softwaresystemen für den Unternehmenseinsatz bewährt. Daher liegt es nahe, das Prinzip der Anwendungsorientierung nicht nur für die Konstruktion einzelner Softwaresysteme anzuwenden, sondern auch auf die Umgestaltung von Anwendungslandschaften. Diese Idee wurde durch die berufliche Tätigkeit des Autors beeinflusst: Er war beratend und aktiv eingreifend an mehreren Projekten beteiligt, die dem Typus „Umgestaltungsprojekt“ entsprechen. In einigen dieser Projekte kam ein anwendungsorientierter Modellierungsansatz, die *exemplarische Geschäftsprozessmodellierung* (siehe Abschnitt 4.2.5) zum Einsatz. Der Autor ist Teil eines Teams, das diesen Modellierungsansatz entwickelt. Er hat sich unter anderem mit der Aufgabe beschäftigt, den Ansatz um Darstellungsmöglichkeiten für Anwendungslandschaften zu erweitern. Dieser praktische Zugang zum Thema ist für diese Arbeit essentiell, denn die Umgestaltung von Anwendungslandschaften wurde bisher nicht hinreichend von anderen Autoren analysiert und kann auf Grund ihrer Komplexität nicht für Versuchszwecke in einem Labor nachgestellt werden.

Neben anwendungsorientierten und betriebswirtschaftlichen Perspektiven werden Anwendungslandschaften in Forschung und Praxis auch aus einer technologischen Perspektive betrachtet. Diese rückt unter anderem *serviceorientierte Architekturen* (SOA, siehe z.B. [BEL⁺09] und [EHH⁺08]) in den Vordergrund. Mit SOA wird eine Flexibilisierung von Anwendungslandschaft angestrebt, die den sich häufig ändernden Geschäftsprozessen Rechnung trägt. Dazu werden Anwendungslandschaften nicht als Verbund von mehr oder weniger gut integrierten, einzelnen Softwaresystemen aufgefasst, sondern als interagierende Services. Die technische Integration

einzelner Softwaresysteme wird auch unter dem Schlagwort *Enterprise Application Integration* (vgl. [HW04]) diskutiert.

Die technologische Perspektive beleuchtet des Weiteren den Betrieb der Softwaresysteme einer Anwendungslandschaft. Die für den Betrieb nötigen Arbeitsabläufe und organisatorischen Strukturen einer IT-Abteilung werden z.B. in der *IT Infrastructure Library (ITIL)* und im *Microsoft Operations Framework* (beide siehe [Kel07]) beschrieben. ITIL fokussiert auf das IT-Servicemanagement, wie beispielsweise Störungs- und Releasemanagement. Um solche Services umzusetzen, benötigt eine IT-Abteilung detaillierte Kenntnisse über den technischen Zustand der betreuten Anwendungslandschaft. Die dafür nötigen Informationen werden typischer Weise in einer *Configuration Management Database (CMDB)*, siehe [Kel07]) gehalten.

Zusammenfassend ordnet sich diese Arbeit in die wissenschaftliche Diskussion von Anwendungslandschaft ein, indem sie bestehenden Perspektiven eine weitere hinzufügt. Betriebswirtschaftliche und technologische Aspekte von Anwendungslandschaften wie beispielsweise Kosten, Wartbarkeit, Sicherheit und Skalierbarkeit sind von großer praktischer Relevanz für Umgestaltungsprojekte:

- Die Managementprozesse der betriebswirtschaftliche Perspektive können Umgestaltungsprojekte auslösen.
- Zur technologischen Perspektive gehört „Handwerkszeug“ für die technische Umsetzung einer geplanten Umgestaltung.
- Die anwendungsorientierte Perspektive ist für Umgestaltungsprojekte wichtig, weil sie die Einsetzbarkeit der umgestalteten Anwendungslandschaft in den Fokus rückt.
- Alle Perspektiven liefern Informationen (etwa in Form von Modellen der Unternehmensarchitektur oder einer CMDB), auf deren Basis Entscheidungen über die Umgestaltung getroffen werden.

Daher müssen bei einer Umgestaltung zusätzlich zur anwendungsorientierten Perspektive auch der Betrieb von Anwendungslandschaften, ihre Architektur und ihre betriebswirtschaftliche Bedeutung berücksichtigt werden.

1.3 Forschungsmethodik

Um eine Methodik für die wissenschaftliche Auseinandersetzung mit der Umgestaltung von Anwendungslandschaften auszuwählen, müssen mehrere Faktoren berücksichtigt werden:

- Komplexe Anwendungslandschaften können nur durch die Kooperation einer Vielzahl von Menschen umgestaltet werden.
- In Umgestaltungsprojekten ändern sich oft Randbedingungen, etwa Termine und der betroffene Ausschnitt der Anwendungslandschaft.
- Modelle müssen dem mit der Zeit zunehmenden Wissen über die Anwendungslandschaft Rechnung tragen und im Projektverlauf wiederholt angepasst werden.

*Design Research*² (vgl. [HMPR04], [MS95], [Win08]) eignet sich als Methodik für Problemfelder mit den oben beschriebenen Eigenschaften. Der folgende Abschnitt gibt eine kurze Einführung in diese Forschungsmethodik.

1.3.1 Design Research

Design Research berücksichtigt „unstable requirements and constraints based upon ill-defined environmental contexts“ [HMPR04, S.81] und „dependence upon human social abilities (e.g., teamwork)“ [HMPR04, S.81]. Die Änderbarkeit von Ergebnissen wie z.B. Modellen wird von Hevner et al. sogar gefordert („flexibility to change design processes as well as design artifacts“ [HMPR04, S.81]).

Daher wurden die in dieser Arbeit beschriebenen Forschungsergebnisse nach der Design-Research-Methodik erarbeitet. In der Terminologie des Design Research werden Ergebnisse *Artefakte* genannt und verschiedene Artefakttypen unterschieden³:

- *Konzepte* (engl. *constructs*) umfassen das Vokabular zur Beschreibung von Problemen und deren Lösungen (vgl. [HMPR04, S.77ff]). Sie bestimmen, wie

²Ich folge der Terminologie Winters [Win08], der die Anwendung der Forschungsmethodik als *Design Research* bezeichnet und ihre Verbesserung als *Design Science*.

³Die von einigen Autoren geforderte Ergänzung um den Typ *Theorie* (siehe [Win08]) nehme ich nicht vor, da keine Ergebnisse dazu erzeugt wurden.

das Problemfeld wahrgenommen und vergegenständlicht wird. Dadurch haben sie großen Einfluss auf den Entwurf anderer Artefakte.

- *Modelle* nutzen das in den Konzepten entwickelte Vokabular, um Situationen aus der Praxis – Probleme und ihren Lösungsraum – darzustellen. Damit helfen Modelle, Probleme, Lösungen und ihren Zusammenhang zu verstehen (vgl. [HMPR04, S.77ff]).
- *Methoden* umfassen Algorithmen und Vorgehensweisen. Methoden bestehen aus einer Menge von Schritten zur Lösung bestimmter Aufgaben (vgl. [MS95]).
- *Ausprägungen* (engl. *instantiations*) können die Brauchbarkeit anderer Artefakte zeigen (vgl. [MS95]). Beispielsweise kann ein Softwaresystem eine Ausprägung eines Modells sein.

Im Design Research entstehen Artefakte in einem zyklischen Prozess, der in folgende Schritte eingeteilt wird (vgl. [MS95]):

- *Konstruieren* (engl. *build*): Artefakte werden mit dem Zweck konstruiert, bestimmte Aufgaben zu lösen.
- *Evaluieren* (engl. *evaluate*): Wie gut sich Artefakte für ihre Zwecke eignen, wird durch diesen Schritt bewertet.
- *Erklären* (engl. *theorize*): Nach der Frage ob ein Artefakt nützlich ist, kann geklärt werden, warum es nützlich ist. Dazu können bestehende Theorien verschiedener wissenschaftlicher Disziplinen herangezogen werden.
- *Verteidigen* (engl. *justify*): Eine aufgestellte Erklärung der Nützlichkeit eines Artefakts wird durch empirische Methoden überprüft.

Abbildung 1.3 stellt die Schritte und Artefakte des Design Research in einer Matrix gegenüber. Die Zellen dieser Matrix repräsentieren mögliche Forschungsergebnisse. Die Beiträge dieser Arbeit wurden in der Matrix in grauer Farbe gekennzeichnet. Diese Arbeit adressiert mehrere Zellen, was im Design Research durchaus üblich ist (vgl. [MS95, S.260]). Die Matrix verdeutlicht, dass die vorliegende Arbeit konzeptueller Natur ist, da der Autor überwiegend Artefakte des Typs „Konzept“ erarbeitet hat. Der folgende Abschnitt beschreibt die Artefakte, wie sie erarbeitet und wie sie evaluiert wurden.

	Konstruieren	Evaluieren	Erklären	Verteidigen
Konzepte				
Modelle				
Methoden				
Ausprägungen				

Abbildung 1.3: Artefakte und Schritte des Design Research

1.3.2 Ergebnisse

Das wichtigste Ergebnis, das in dieser Arbeit konstruiert wird, sind Bausteine zur Anpassung von Modellierungsansätzen an Umgestaltungsprojekte. Die in den Bausteinen verarbeiteten Konzepte dienen dazu, in der Praxis vorkommende Modellierungsprobleme in Umgestaltungsprojekten zu minimieren. Da „Umgestaltungsprojekt“ noch nicht als Projekttyp in der Informatik etabliert ist, steht vor der Konstruktion der Bausteine ein weiterer Konstruktionsschritt. Dieser produziert Konzepte, mit denen die Rolle der Modellierung bei der Umgestaltung von Anwendungslandschaften analysiert werden kann.

Einige Bausteine helfen dabei, Artefakte vom Typ „Modell“ zu entwickeln, weil sie ein Metamodell und Modelltypen beschreiben.

Die Konstruktion von Methoden ist insofern Thema dieser Arbeit, da mit den Bausteinen beispielhaft ein existierender Modellierungsansatz angepasst wird. Dieses Anwendungsbeispiel gibt einige Anhaltspunkte für die Entwicklung einer Entwurfsmethode für Modellierungsansätze. Es ist jedoch nicht Ziel dieser Arbeit, eine solche zu beschreiben.

Design Research ist motiviert durch das Ziel, eine Verbesserung der Praxis herbeizuführen [MS95, S.251]. Um evaluieren zu können, ob mit den Bausteinen tatsächlich Modellierungsansätze besser an Umgestaltungsprojekte angepasst werden können, werden Qualitätsmerkmale für die Bausteine festgelegt:

- Sie müssen *relevant* sein, d.h. die Modellierungsprobleme minimieren.
- Sie müssen *anwendbar* sein, d.h. mit den Bausteinen muss ein Modellierungsansatz angepasst werden können.
- Sie müssen *praxistauglich* sein, d.h. ein mit den Bausteinen angepasster Ansatz muss sich für den Unternehmenseinsatz eignen.

Design Research lässt offen, wie die Zielerreichung evaluiert wird. In dieser Arbeit werden dazu mehrere Indikatoren eingesetzt. Zu diesen Indikatoren zählen unter anderem:

- Ein werkzeuggestützter und um die Bausteine ergänzter Modellierungsansatz.
- Ein Anforderungskatalog, mit dem der entwickelte Ansatz und weitere, praxisrelevante Modellierungsansätze bewertet werden.
- Die ex-post Modellierung eines realen Umgestaltungsprojekts mit dem angepassten Ansatz.

Ein weiterer möglicher Indikator ist der Praxiseinsatz des entwickelten Modellierungsansatzes. Die Rahmenbedingungen dafür waren jedoch nicht gegeben.

Neben der Konstruktion und Evaluation von Artefakten wurde für diese Arbeit noch ein dritter Schritt des Design-Research-Zyklus umgesetzt: Die empirisch festgestellten Modellierungsprobleme werden durch Theorie zu Anwendungsorientierung, Modellierung und Software-Technik erklärt. Die daraus gewonnenen Erkenntnisse gingen in die Konstruktion der Bausteine ein.

Zusammengefasst liegt der Fokus dieser Arbeit auf der Konstruktion und Evaluation von . . .

- Konzepten, die „Umgestaltung von Anwendungslandschaften“ und „Umgestaltungsprojekte“ einführen,
- Konzepten und Modellen, welche Bausteine zur Anpassung von Modellierungsansätzen an Umgestaltungsprojekte beschreiben und
- einer Ausprägung des Bausteine in Form eines werkzeuggestützten Modellierungsansatzes.

Die folgenden Absätze beschreiben den Prozess, der zu diesen Ergebnissen geführt hat. Um den Lesefluss der Arbeit zu verbessern, wird in anderen Kapiteln nicht mehr detailliert auf den Prozess eingegangen, sondern auf diesen Abschnitt verwiesen. Der Prozess gliedert sich grob in drei Schritte (siehe Abbildung 1.4):

Im ersten Schritt wird untersucht, was ein Umgestaltungsprojekt ausmacht und ob dieser Projekttyp praktische Relevanz hat. Dazu wurden in einer empirischen Untersuchung sieben „Umgestaltungsszenarien“ aufgestellt (siehe Anhang A.1) und

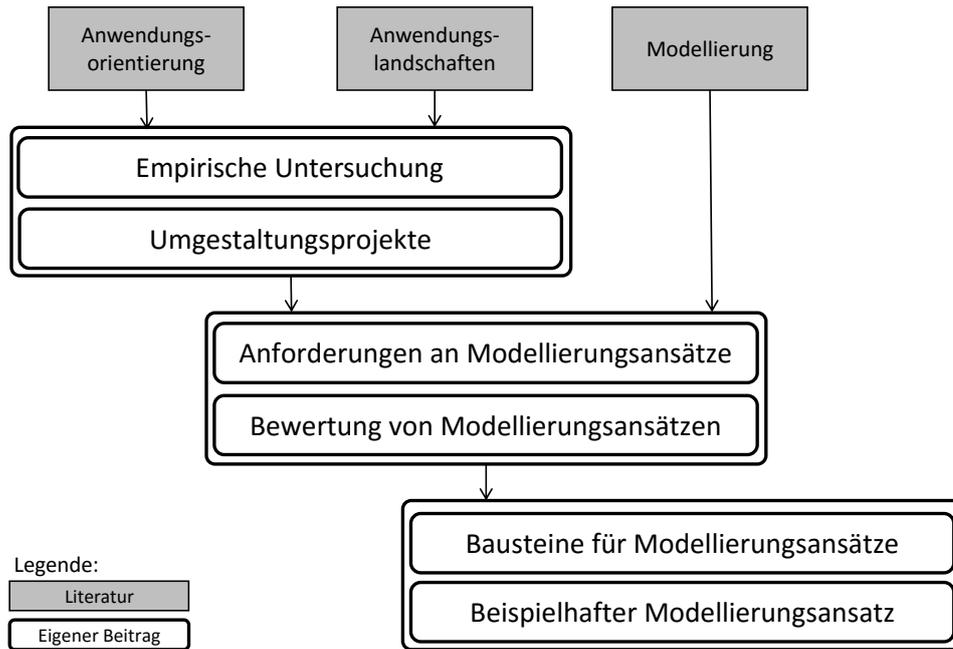


Abbildung 1.4: Vorgehen bei der Ausarbeitung der Ergebnisse

von Experten bewertet. Basis für die Szenarien waren Literaturquellen ([EHH⁺08], [MW04]) und vier Praxisprojekte, die der Arbeitgeber des Autors zwischen 2007 und 2010 in den Branchen Finanzdienstleistung, Logistik und Großhandel durchgeführt hatte. Die in diesen Projekten geleisteten Aufwände reichen von wenigen tausend bis zu mehreren zehntausend Personentagen. An drei der Projekte war der Autor dieser Arbeit beteiligt.

Die Szenarien wurden in halbstrukturierten Interviews durch fünf Experten bewertet, die als Softwareentwickler, Softwarearchitekt, Projektleiter und Berater an den untersuchten oder vergleichbaren Projekten beteiligt waren. „Umgestaltung“ wurde von den Interviewten als relevanter Projekttyp bewertet (siehe Anhang A.1) und sie identifizierten mögliche Probleme in solchen Projekten. Jene Probleme, die mit Hilfe von Modellen lösbar schienen, wurden vom Autor zusammengefasst. Ein weiteres Ergebnis der Interviews war eine Liste von typischen Umgestaltungstätigkeiten.

Um zu bewerten, wie relevant die Modellierungsprobleme und Umgestaltungstätigkeiten sind, wurden diese beiden Ergebnisse in Form eines Fragebogens sieben weitere Experten vorgelegt. Die befragten Personen waren in unterschiedlichen Rollen (Projektleiter, IT-Abteilungsleiter, Berater und Software Architekt) an Umgestaltungsprojekten beteiligt und es gab keine Überschneidungen mit den Teilnehmern

der Interviews. Auf die als relevant bewerteten Umgestaltungstätigkeiten und Modellierungsprobleme wird in Kapitel 2 detailliert eingegangen.

Für den zweiten und dritten Schritt wurde auf eine Modellierungsaufgabe aus einem realen Umgestaltungsprojekt zurückgegriffen. Ein Projektmitarbeiter verfasste eine Diplomarbeit (siehe [Ros09]) über das Projekt und beschreibt es darin so ausführlich, dass dafür ex-post Modelle erstellt werden konnten.

Im zweiten Schritt wurde die Frage behandelt, wie gut sich existierende Modellierungsansätze für die Modellierung in Umgestaltungsprojekten eignen. Dazu wurde ein Anforderungskatalog aufgestellt (siehe Abschnitt 3.5) und fünf zu bewertende Modellierungsansätze ausgewählt (siehe Abschnitt 4.1). Bewertet wurden die Ansätze anhand von Literaturquellen und in dem der Autor die Ansätze zur Lösung der Modellierungsaufgabe einsetzte. Im Lauf der Bewertung wurde der Anforderungskatalog so lange ergänzt und präzisiert, bis sich ein differenziertes Bild der Ansätze ergab. Dies war nach 13 Anforderungen der Fall.

Das Ziel des dritten Schritts war, Bausteine zu konstruieren, mit denen existierende Ansätze an Umgestaltungsprojekte angepasst werden können. Dazu wurde jeder der zuvor bewerteten Ansätze darauf untersucht, mit welchen Sprachmitteln und Vorgehensweisen er Anforderungen erfüllt. Für alle Anforderungen bis auf eine⁴ konnten passende Sprachmittel und Vorgehensweisen identifiziert werden. Der Autor generalisierte sie soweit, dass sie auch außerhalb ihres ursprünglichen Ansatzes als „Blaupause“ verwendet werden konnten. Die insgesamt zehn⁵ generalisierten Sprachmittel und Vorgehensweisen sind die gesuchten Bausteine. Sie wurden parallel abgeleitet und in einem werkzeuggestützten Modellierungsansatz implementiert. So konnte sichergestellt werden, dass die Bausteine generisch genug formuliert waren, um in andere Ansätze integriert werden zu können. Weil die Bausteine das wichtigste Ergebnis dieser Arbeit darstellen, ist ihrer Evaluation das ganze Kapitel 6 gewidmet. Unter anderem wurde die reale Modellierungsaufgabe mit dem um die Bausteine erweiterten Modellierungsansatz gelöst und die Modelle dem Projektmitarbeiter vorgelegt, der das Projekt in seiner Diplomarbeit beschrieben hatte (siehe [Ros09]). Anhand eines Fragebogens wurde evaluiert, ob die ex-post erstellten Modelle eine Verbesserung gegenüber der damals genutzten Modelle darstellen.

⁴Für eine Anforderung zur Modellmanipulation und -analyse ließ sich – abstrahiert von konkreten Modellierungssprachen und -werkzeugen – kein Baustein ableiten.

⁵Einer der Bausteine adressiert drei Anforderungen, daher kommen zehn Bausteine auf 12 Anforderungen.

1.4 Aufbau der Arbeit

Der Aufbau dieser Arbeit folgt dem in Abbildung 1.4 dargestellten Vorgehen:

Zu Beginn wird in Kapitel 2 eine anwendungsorientierte Perspektive auf Anwendungslandschaften eingeführt und die Umgestaltung von Anwendungslandschaften motiviert. Mittel zum Zweck der Umgestaltung sind Umgestaltungsprojekte. Diese werden als Projekttyp definiert. Damit wird in diesem Kapitel die erste Forschungsfrage (*Was charakterisiert Umgestaltungsprojekte?*) beantwortet.

In Kapitel 3 wird die Bedeutung von Modellen für Umgestaltungsprojekte dargestellt und typische Modellierungsprobleme in solchen Projekten beschrieben. Zwei weitere Forschungsfragen (*Zweck von Modellen, Modellierungsprobleme*) werden damit beantwortet. Außerdem werden in Kapitel 3 die grundlegenden Eigenschaften der Modellierung in Umgestaltungsprojekten herausgearbeitet. Auf ihnen bauen die weiteren Ergebnisse dieser Arbeit auf, weil die grundlegenden Eigenschaften die Forschungsfrage beantworten, welche Gründe es für die Modellierungsprobleme gibt. Um überprüfbar zu machen, wie stark die Eigenschaften in existierenden Ansätzen ausgeprägt sind, wird ein Anforderungskatalog für Modellierungsansätze aufgestellt.

In Kapitel 4 wird bewertet, in wie weit gängige Ansätze die Anforderungen erfüllen. Für die Bewertung werden fünf praxisrelevante Modellierungsansätze ausgewählt.

Kapitel 5 beantwortet die letzte Forschungsfrage, indem untersucht wird, mit welchen Mitteln die zuvor bewerteten Ansätze die Anforderungen umsetzen. Diese Mittel, wie z.B. bestimmte Eigenschaften der Modellierungssprache, werden zu „Bausteinen“ verallgemeinert. Diese Bausteine ermöglichen, Modellierungsansätze so anzupassen, dass sie die Anforderungen aus Kapitel 3 erfüllen. Kapitel 5 zeigt auch die Anwendung der Bausteine. Dazu wird mit Hilfe der Bausteine ein bestehender Modellierungsansatz besser an den Einsatz in Umgestaltungsprojekten angepasst. Ob der dadurch entstandene Ansatz die Anforderungen an Modellierungsansätze für Umgestaltungsprojekte besser erfüllt als die zuvor untersuchten Ansätze, wird in Kapitel 6 bewertet.

Abschließend werden in Kapitel 7 Anknüpfungspunkte für eine weitere wissenschaftliche Beschäftigung mit der Umgestaltung von Anwendungslandschaften aufgezeigt.

Kapitel 2

Umgestaltung von Anwendungslandschaften

Ziel dieses Kapitels ist, die Notwendigkeit der Umgestaltung von Anwendungslandschaften zu zeigen und damit den Gegenstandsbereich dieser Arbeit vorzustellen. Dazu werden „Anwendungslandschaft“, „Umgestaltung von Anwendungslandschaften“ und der Projekttyp „Umgestaltungsprojekt“ definiert. Auf Grundlage dieser Begriffe kann im folgenden Kapitel 3 diskutiert werden, welchen Zweck die Modellierung in Umgestaltungsprojekten erfüllen soll.

2.1 Anwendungslandschaften

Anwendungslandschaft ist ein in der Literatur häufig verwendeter, aber verschieden definierter Begriff. Nach Matthes und Wittenburg umfasst er „die Gesamtheit der betrieblichen Informationssysteme in einem Unternehmen“ [MW04, S.4]), wobei solche Informationssysteme laut Dern „zur Unterstützung der Durchführung von Geschäftsprozessen betrieben werden“ [Der03, S.16]. Oft werden neben den Informationssystemen selbst auch noch ihre *Abhängigkeiten* in die Definition eingeschlossen (z.B. in [EHH⁺08] und [Sta15]). Diese werden aber ebenfalls verschieden definiert. Einige Autoren verstehen darunter nur die Beziehungen der Softwaresysteme untereinander. Andere wie beispielsweise Buckl et al. fassen unter anderem auch die Beziehungen der Softwaresysteme zu den Geschäftsprozessen eines Unternehmens als Abhängigkeiten auf. (vgl. [BEMS09]). In der Literatur finden sich diverse, zu „Anwendungslandschaft“ synonym verwendete Begriffe: *Applikationslandschaft*

(z.B. in [Sch05]), *Informationssystemlandschaft* (z.B. in [Han13]) und *application landscape* (z.B. in [BELM08], [Lan13] und [Sch09]). Diese Begriffe werden im Rahmen des oben beschriebenen Interpretationsspielraums verwendet.

Wie bereits die Beispiele in der Einleitung von Kapitel 1 zeigen, erfordert die Umgestaltung von Anwendungslandschaften, diese im Kontext ihres Einsatzes in Geschäftsprozessen zu betrachten. Eine solche anwendungsorientierte Perspektive (siehe Abschnitt 1.2) fehlt den oben genannten Begriffsdefinitionen, weil darin unter anderem keine Anwender berücksichtigt werden. Daher wird im Folgenden eine eigene Definition von „Anwendungslandschaft“ erarbeitet, welche nicht nur eine technologische und betriebswirtschaftliche, sondern auch eine anwendungsorientierte Perspektive berücksichtigt. Die für die Definition benötigten Begriffe werden schrittweise eingeführt und ihre Zusammenhänge durch Abbildungen verdeutlicht. Am Ende des Abschnitts werden die Abbildungen in eine Grafik zusammengefügt, welche eine anwendungsorientierte Perspektive auf Anwendungslandschaften veranschaulicht.

Die in dieser Arbeit betrachteten Anwendungslandschaften sollen einen Beitrag zur Umsetzung von Geschäftsprozessen¹ leisten:

Geschäftsprozess (Definition 2.1.1): „Ein Geschäftsprozess besteht aus einer zusammenhängenden, abgeschlossenen Folge von Tätigkeiten, die zur Erfüllung einer betrieblichen *Aufgabe* notwendig sind.“ [Sta06, S.9]

Geschäftsprozesse setzen also Aufgaben um:

Aufgabe (Definition 2.1.2): „Eine Aufgabe ist eine betriebliche Funktion mit einem bestimmbar Ergebnis. Sie wird von Menschen und/oder Maschinen ausgeführt.“ [Oes95, S.50]

Die ausführenden Personen haben in Bezug auf die Aufgabe eine bestimmte Rolle und werden im Folgenden *Akteure* genannt (vgl. [Rol98]). Mehrere, zu einer Aufgabe gehörende menschliche Tätigkeiten werden in dieser Arbeit als *Arbeitsablauf* bezeichnet. Ein und derselbe Geschäftsprozess kann, z.B. in Abhängigkeit von den Akteuren, durch verschiedene konkrete Arbeitsabläufe umgesetzt werden. Abbildung 2.1 zeigt die bisher eingeführten Begriffe und ihre Zusammenhänge.

¹Folgende Definition wurde gewählt, weil sie aus fünf anderen Definitionen und weiterer Literatur hergeleitet wurde und damit fundiert ist.

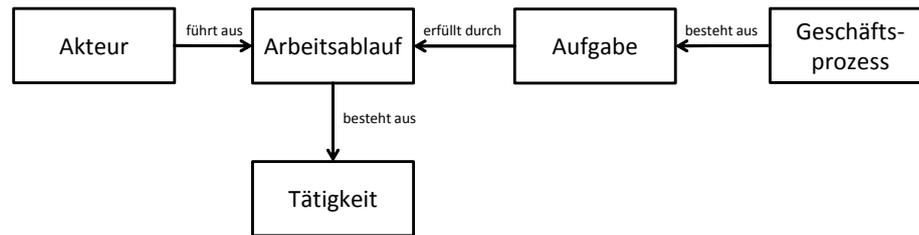


Abbildung 2.1: Begriffe und ihr Zusammenhang (Teil 1)

Akteure führen Tätigkeiten aus, um bestimmte Arbeitsergebnisse (z.B. Dokumente oder physische Erzeugnisse) zu erzielen. Dazu benötigen sie gegebenenfalls Arbeitsgegenstände wie beispielsweise Materialien, Kommunikationsmedien und Zustandsinformationen über Arbeitsergebnisse. Arbeitsgegenstände und -ergebnisse stehen daher miteinander in Beziehung. Aber auch Arbeitsergebnisse können miteinander in Beziehung stehen (z.B. wenn zu einem *Antrag* ein *Vertrag* existiert).

In der Regel verrichten Akteure in Unternehmen *kooperative Arbeit*, bei der sie „geplant und koordiniert zusammen [arbeiten], um ein gemeinsames Ergebnis zu erreichen.“ [Zül98, S. 428]. Dazu müssen die Akteure miteinander kommunizieren und sich koordinieren. Daher gibt es in Arbeitsabläufen nicht nur Tätigkeiten, in denen Arbeitsergebnisse bearbeitet werden, sondern auch Tätigkeiten, die der Koordination der Arbeitsabläufe dienen. In letzteren werden Informationen über die Arbeitsabläufe benötigt (z.B. in Form eines Arbeitsgegenstands *Vorgangsmappe*), die Akteuren untereinander austauschen. Arbeitsgegenstände und -ergebnisse werden in kooperativen Arbeitsabläufen ebenfalls von Akteur zu Akteur weitergegeben. Werkzeuge unterstützen die Akteure in ihren Tätigkeiten, in dem sie ermöglichen, Arbeitsgegenstände und -ergebnisse zu bearbeiten, sie auszutauschen und Informationen über die Arbeitsabläufe zu kommunizieren. Abbildung 2.2 stellt diese Zusammenhänge dar.

Informationen über Arbeitsabläufe, Arbeitsgegenstände und Arbeitsergebnisse können in Form von digitalen Daten durch Softwaresysteme realisiert werden. Softwaresysteme stellen ihren Benutzern Funktionen bereit, um digitale Arbeitsgegenstände und digitale Arbeitsergebnisse zu bearbeiten und um digital zu kooperieren. Softwaresysteme realisieren also „Software-Werkzeuge“ (siehe Abbildung 2.3) und sind damit Arbeitsmittel für menschliche Tätigkeiten. Wenn in einem Unternehmen die IT-Unterstützung von Geschäftsprozessen analysiert werden soll, muss betrachtet werden, wie die im Prozess zu erfüllenden Aufgaben durch IT-gestützte Tätigkeiten

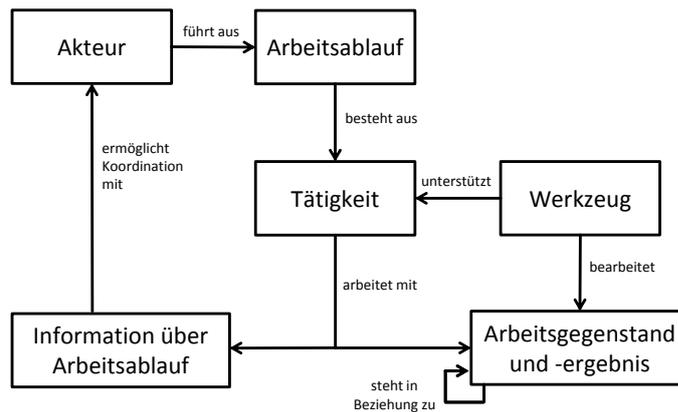


Abbildung 2.2: Begriffe und ihr Zusammenhang (Teil 2)

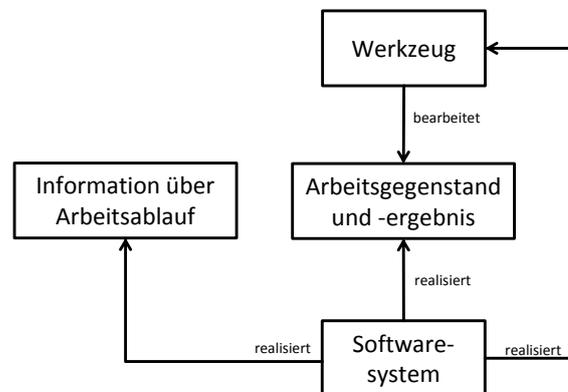


Abbildung 2.3: Begriffe und ihr Zusammenhang (Teil 3)

ausgeführt werden.

Damit kann zusammengefasst werden, dass die in dieser Arbeit betrachteten Anwendungslandschaften in einen menschlichen Arbeitskontext eingebettet sind. Zwar können Softwaresysteme darin als *technische Akteure*² auftreten und eigenständig Aufgaben erledigen, doch wird zumindest ein Teil der Softwaresysteme einer Anwendungslandschaft von menschlichen Anwendern direkt genutzt. Das begründet auch die Wahl des Begriffs „Anwendungslandschaft“ statt beispielsweise „Systemlandschaft“. Daher wird z.B. ein Verbund von Softwaresystemen, die zusammen automatisiert Geschäftsprozesse erledigen, in dieser Arbeit nicht als „Anwendungslandschaft“ bezeichnet. Anwendungslandschaften sind also *Mensch/Aufgabe/Technik-Systeme*:

²Im Folgenden werden technische Akteure immer explizit als solche bezeichnet. Sind handelnde Personen oder Personen *und* Softwaresysteme gemeint, wird der Begriff „Akteur“ verwendet.

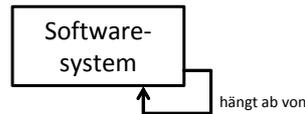


Abbildung 2.4: Begriffe und ihr Zusammenhang (Teil 4)

Mensch/Aufgabe/Technik-System, MAT-System (Definition 2.1.3): In *MAT-Systemen* beeinflussen sich Mensch, Aufgabe und Technik gegenseitig. „Technik als Element von Informations- und Kommunikationssystemen ist Hardware, Software und deren Zusammenwirken“ [Hei02, S.15]. In MAT-Systemen dient Technik zur Erledigung von Aufgaben, wobei der Mensch die Technik steuert (vgl. [Hei02, S.14]). „Der Mensch darf daher nicht als 'Human-Peripherie' oder als 'Lückenbüßer' der Technik verstanden werden.“ [Hei02, S.14]

Als MAT-Systeme haben Anwendungslandschaften den Zweck, Menschen in ihren Aufgaben zu unterstützen.

Neben ihrem Einsatz in Geschäftsprozessen weisen Anwendungslandschaften noch ein zweites, technisches Merkmal auf. Mehrere Softwaresysteme werden nur dann als Anwendungslandschaft bezeichnet, wenn sie zumindest teilweise voneinander abhängig sind (siehe Abbildung 2.4):

Abhängigkeit (Definition 2.1.4): Jede Art von Kopplung zwischen Softwaresystemen. Eine Abhängigkeit kann sowohl zwischen Softwaresystemen derselben oder einer anderen Organisation bestehen.

Technisch können Softwaresysteme über verschiedene Mechanismen gekoppelt werden. Beispielsweise werden in [HW04]) folgende erläutert:

- **Messaging:** Softwaresysteme tauschen Daten über ein gemeinsames Nachrichtensystem aus.
- **Remote Procedure Invocation:** Ein Softwaresystem ruft eine Funktion eines anderen Softwaresystems auf.
- **Shared Database:** Softwaresysteme tauschen Daten über eine gemeinsame Datenbank aus.
- **File Transfer:** Ein Softwaresystem schreibt Daten in eine Datei, die von einem anderen Softwaresystem verarbeitet wird.

Starke führt in [Sta15, S.65f] noch weitere Arten von Kopplung auf, die über den Aufruf von Funktionen und gemeinsam genutzte Daten oder Datenstrukturen hinausgehen. Diese Kopplungsarten weisen nicht unbedingt eine technische Entsprechung in Form einer Schnittstelle auf:

- Kopplung über Hardware (z.B. gleiches Netzwerksegment) oder Laufzeitumgebung (z.B. geteilter Adressraum, selbe virtuellen Maschine).
- Kopplung über die Zeit (d.h. Aktivitäten verschiedener Softwaresysteme müssen in einer bestimmten Reihenfolge erfolgen).

Damit wurden alle Elemente für eine anwendungsorientierte Definition des Begriffs *Anwendungslandschaft* hergeleitet:

Anwendungslandschaft (Definition 2.1.5): Eine *Anwendungslandschaft* ist ein Mensch/Aufgabe/Technik-System, das von einer Organisation zur Unterstützung ihrer Geschäftsprozesse eingesetzt wird. Eine Anwendungslandschaft umfasst die zu diesem Zweck betriebenen Softwaresysteme der Organisation, sowie die Abhängigkeiten zwischen diesen Softwaresystemen und anderen Softwaresystemen.

Abbildung 2.5 stellt die in diesem Abschnitt beschriebenen Begriffe und Zusammenhänge dar und beschreibt damit eine anwendungsorientierte Perspektive auf Anwendungslandschaft.

2.2 Umgestaltungsnotwendigkeiten und Ziele

Eine Anwendungslandschaft umzugestalten kann aus verschiedenen Gründen notwendig sein (vgl. [Kel07]):

- Gesetzliche Vorgaben wie z.B. die Einführung des *einheitlichen Euro-Zahlungsverkehrsraums (SEPA)*.
- Geschäftliche Entscheidungen wie z.B. Fusion von Unternehmen und der Wunsch nach effizienteren Geschäftsprozessen.
- Technische Gründe wie z.B. die oft schlechte Wartbarkeit jahrzehntealter Softwaresysteme.

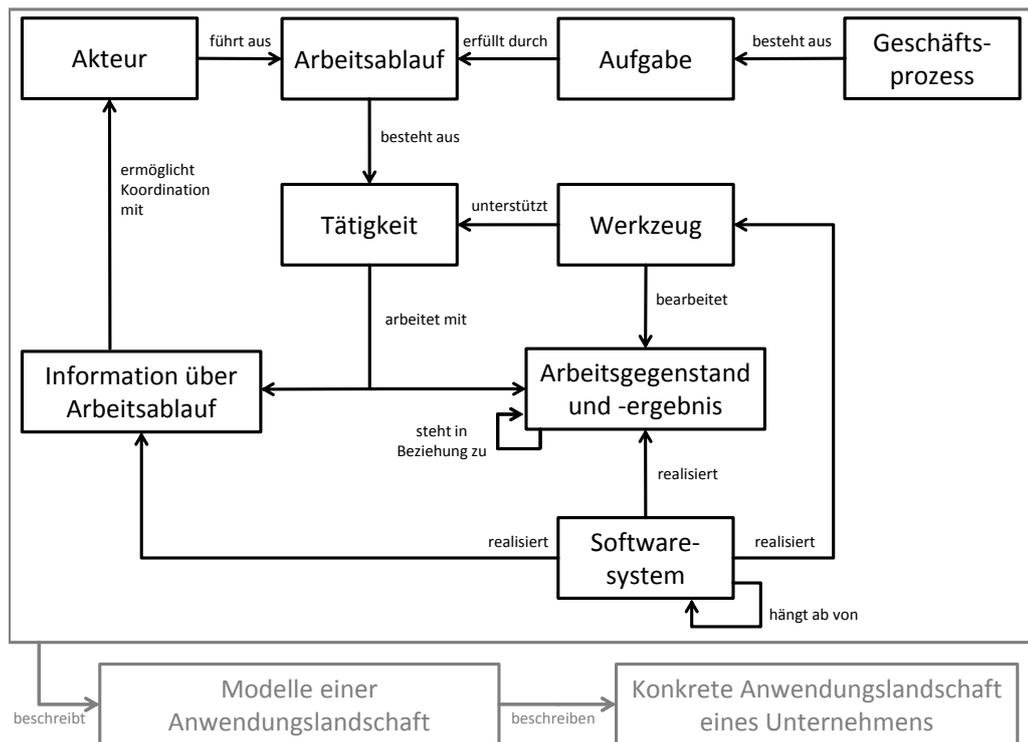


Abbildung 2.5: Anwendungsorientierte Perspektive auf Anwendungslandschaften

Die Ziele, die mit einer Umgestaltung erreicht werden sollen, sind vielfältig: Kosten sollen eingespart, gesetzliche Vorgaben eingehalten, Geschäftsprozesse besser unterstützt werden und Vieles mehr. Konkreter Auslöser einer Umgestaltung kann z.B. eine strategische Entscheidung sein, die im Rahmen des Enterprise Architecture Management (siehe Abschnitt 1.2) eines Unternehmens getroffen wurde.

Unabhängig von ihrem Auslöser gibt es in Projekten allgemeine Ziele: Budgets und Zeitpläne sollen eingehalten, Aufgaben erledigt und Fehler vermieden werden. Im Kontext von Umgestaltung bedeutet das Vermeiden von Fehlern, dass die von der Anwendungslandschaft unterstützten Geschäftsprozesse während und nach der Umgestaltung ausgeführt werden können. Je schwerer die Auswirkung einer Änderung der Anwendungslandschaft einzuschätzen ist, desto höher ist das damit verbundene Risiko. Sind die Zusammenhänge zwischen Geschäftsprozessen und Anwendungslandschaft oder die Abhängigkeiten innerhalb der Anwendungslandschaft komplex, werden Fehler wahrscheinlicher. Folgendes Beispiel (vgl. [Hof09]) soll einen Eindruck davon geben, welche Überlegungen nötig sind, um die Risiken einer Umgestaltung abzuschätzen:

Eine Bank wechselt aus technischen Gründen das Betriebssystem aller Client-Server-Anwendungen und muss deswegen ihre Anwendungslandschaft anpassen:

- Von der Bank selbst entwickelte Softwaresysteme werden so geändert, dass sie auf dem neuen Betriebssystem ausgeführt werden können.
- Zugekaufte, zum neuen Betriebssystem inkompatible Softwaresysteme werden durch vergleichbare, kompatible Softwaresysteme ersetzt.
- Einige Systeme werden außer Betrieb genommen, weil ihre Funktionalität zukünftig als externe Dienstleistung zugekauft wird.

Eines der Softwaresysteme, das außer Betrieb genommen wird, dient zur Abwicklung des Wertpapierhandels. Auf Grund seiner wirtschaftlichen Bedeutung und der tiefen Integration in die bestehende Anwendungslandschaft ist die Außerbetriebnahme mit großen Risiken für die Bank verbunden. Um diese handhabbar zu machen, müssen unter anderem folgende Fragen geklärt werden:

- Welcher Ausschnitt der Anwendungslandschaft ist von den Veränderungen im Wertpapierhandel betroffen?
- Welche Softwaresysteme benötigen Daten, die bislang im Wertpapierhandelssystem gehalten wurden?
- Wie wirken die IT-gestützten und manuellen Arbeitsabläufe zusammen, die den Geschäftsprozess „Wertpapierhandel“ ausmachen?
- Welche Geschäftsprozesse müssen ausgeführt werden, um die an der Anwendungslandschaft durchgeführten Veränderungen zu testen?

Wenn Unternehmen wie die in diesem Beispiel geschilderte Bank ihre Anwendungslandschaften umgestalten, werden sie dazu in der Regel ein Projekt durchführen. Im nächsten Abschnitt wird ein eigener Typ für solche Projekte eingeführt.

2.3 Umgestaltungsprojekte

Bisher wurde „Umgestaltung“ synonym zu „Veränderung“ verwendet. Der Begriff soll in dieser Arbeit jedoch für eine bestimmte Art von Veränderung stehen, die in den folgenden Abschnitten beschreiben wird.

Bei einer Umgestaltung stehen technische Änderungen in Wechselwirkung mit den Geschäftsprozessen und der Struktur einer Organisation. Dies wird auch als *Co-Evolution* bezeichnet:

„The evolution of one domain is partially dependent on the evolution of the other [or] one domain changes in the context of the other.“ [MKP00, S.4]

In dieser Arbeit bezeichnet *Umgestaltung* eine Co-Evolution des technischen Aufbaus und des Einsatzes einer Anwendungslandschaft:

Umgestaltung (Definition 2.3.1): *Umgestaltung* verfolgt einen bestimmten Zweck und verändert dazu zielgerichtet

- die Abhängigkeiten zwischen den Softwaresystemen einer Anwendungslandschaft und
- die Arbeitsabläufe der Akteure, welche die Anwendungslandschaft einsetzen.

Die technischen Änderungen einer Umgestaltung lassen sich auf einige grundlegende Manipulationen zurückführen, mit denen Anwendungslandschaften umgestaltet werden können. Keller beschreibt in [Kel07] mehrere Möglichkeiten, um Anwendungslandschaften zu verändern und er teilt diese in die Kategorien *Create*, *Update* und *Delete* ein (siehe [Kel07, S.95]). Da Keller dabei auch Veränderungen betrachtet, die nicht in den Kontext von Umgestaltung gehören (sondern z.B. zur Wartung), orientiert sich diese Arbeit lediglich an Kellers Kategorisierung und unterscheidet folgende grundlegende Manipulationen an Anwendungslandschaften:

- Inbetriebnahme eines Softwaresystems
- Veränderung des „beobachtbaren Verhaltens“³ eines Softwaresystems

³Die Einschränkung grenzt „Veränderung“ von *Refactorings* [Fow00] ab, die per Definition das beobachtbare Verhalten eines Softwaresystems *nicht* verändern (vgl. [Fow00, S.XVI]). Was das beobachtbare Verhalten eines Softwaresystems ausmacht, hängt vom Einsatzkontext des Softwaresystems ab.

- Außerbetriebnahme eines Softwaresystems

Nur wenn diese Manipulationen mehrere Softwaresysteme – und damit die Abhängigkeiten innerhalb der Anwendungslandschaft – betreffen, handelt es sich um „Umgestaltung“. Bei Inbetriebnahme oder Außerbetriebnahme eines Softwaresystems ist dies der Fall, wenn dadurch neue Abhängigkeiten in der Anwendungslandschaft entstehen oder Abhängigkeiten wegfallen. Zieht eine Veränderung eines Softwaresystems eine Veränderung seiner Abhängigkeiten zu anderen Softwaresystemen nach sich, kann ebenfalls von „Umgestaltung“ gesprochen werden. Entsprechend Definition 2.1.5 können Abhängigkeiten nicht nur betroffen sein, wenn sich Funktionsaufrufe oder ausgetauschte Datenstrukturen ändern, sondern auch, wenn sich zum Beispiel die Laufzeitumgebung verändert oder die Verarbeitungslogik einer Anwendung durch „Kopplung über die Zeit“ andere Softwaresysteme beeinflusst.

Die Manipulationen können in verschiedenen Kombinationen auftreten. Beispielsweise werden Softwaresysteme häufig außer Betrieb genommen und durch andere ersetzt. Solche Vorgänge werden oft als *Migration* bezeichnet (siehe z.B. [LL13]).

Umgestaltung wird in Form von Projekten durchgeführt. Da diese Art von Projekten für diese Arbeit von großer Bedeutung ist, wird ein eigener Begriff dafür eingeführt:

Umgestaltungsprojekt (Definition 2.3.2): Ein *Umgestaltungsprojekt* ist ein zeitlich begrenztes Vorhaben, in dem eine existierende Anwendungslandschaft umgestaltet wird, um einen gewünschten Soll-Zustand zu erreichen.

Auf Grund ihrer Tragweite sind Umgestaltungsprojekte in der Regel unternehmenskritisch. Ihr Scheitern würde die Geschäftsfähigkeit des betroffenen Unternehmens gefährden. Außerdem sind sie so umfangreich, dass sie arbeitsteilig durchgeführt werden.

Bei Umgestaltungsprojekten handelt es sich nicht um eine Sonderform von Softwareentwicklungsprojekten. Bestehende Software zu verändern und neue zu entwickeln kann zwar im Rahmen eines Umgestaltungsprojekts notwendig sein, doch gibt es noch weitere Wege, um eine Anwendungslandschaft zu verändern. Dazu gehört das *Outsourcing* von Geschäftsprozessen und ihrer IT-Unterstützung an Dienstleister. Die Inbetriebnahme einer zugekauften Standard-Software stellt eine weitere Möglichkeit dar, eine Anwendungslandschaft umzugestalten, ohne ein Softwareentwicklungsprojekt durchzuführen.

Die an Umgestaltungsprojekten beteiligten Personen haben eine Reihe typischer Aufgaben und müssen daher verschiedene Tätigkeiten durchführen. Diese Aufgaben und Tätigkeiten werden im Folgenden vorgestellt. Hergeleitet wurden sie mit Hilfe mehrerer Szenarien, die typische Fälle von Umgestaltung beschreiben (siehe Abschnitt 1.3). Alle in dieser Arbeit beschriebenen Aufgaben und Tätigkeiten wurden als relevant für Umgestaltungsprojekte evaluiert. Der zur Evaluation verwendete Fragebogen und die Auswertung der Ergebnisse finden sich im Anhang A.2. Ob die Tätigkeiten einmalig, mehrmals oder regelmäßig ausgeführt werden, ist eine Frage der Ablauforganisation und war nicht Teil der Evaluation.

Informationen erfassen: Die technischen Aspekte (z.B. Abhängigkeiten) und fachlichen Aspekte (z.B. unterstützte Tätigkeiten) der Softwaresysteme, aus denen die Anwendungslandschaft besteht, müssen erfasst werden.

Bewerten und Entscheiden: Das Ziel der Umgestaltung wird bestimmt. Dazu kann die Anwendungslandschaft z.B. hinsichtlich strategischer Vorgaben oder Qualitätsmerkmalen bewertet werden. Ein Beispiel für Qualitätskriterien liefert die Norm *DIN ISO/IEC 25000 (Qualitätskriterien und Bewertung von Softwareprodukten*, siehe [ISO14]). Entscheidungen über Veränderung, Inbetriebnahme und Außerbetriebnahme von Softwaresystemen werden getroffen und der konkrete Handlungsbedarf für einzelne Softwaresysteme und Geschäftsprozesse ermittelt.

Planen: Hierunter fallen viele allgemeine, planerische Aktivitäten, die nicht speziell auf den Gegenstandsbereich „Anwendungslandschaft“ zurückzuführen sind, beispielsweise:

- Projektpläne erstellen.
- Maßnahmen in eine Reihenfolge bringen und in Etappen einteilen.
- Zeitpunkte für die Produktivstellung der Etappen bestimmen.
- Testpläne aufstellen, die auf die Etappen des Umgestaltungsprojekts abgestimmt sind.

Einbinden der Organisation: Hierunter fallen viele kommunikative Tätigkeiten. Sie dienen unter anderem dazu, fachliche und technische Experten in einen Dialog zu bringen, die inhaltlichen Entscheidungen verständlich zu machen und die zeitliche Planung abzustimmen.

- Strategischen Vorgaben für die Umgestaltung ausarbeiten und kommunizieren.
- Zustände und Veränderungen von Anwendungslandschaft und Arbeitsabläufen in der Organisation abstimmen und kommunizieren.
- Rahmenbedingungen für Tests und die Veränderungen an den Arbeitsabläufen an die Tester kommunizieren.
- Projektmitarbeitern erklären, in welchem Kontext sie ihre Aufgaben erledigen, bei Missverständnissen vermitteln und unterschiedliche Sichtweisen aufzeigen.

Technische Umsetzung: Dazu gehört beispielsweise:

- Technische Integration von Softwaresystemen (siehe *Enterprise Application Integration* in Abschnitt 1.2).
- Anwendungen einzeln testen.
- Arbeitsabläufe anwendungsübergreifend testen.

Weil der Fokus dieser Arbeit auf Modellierung liegt, umfasst die Liste überwiegend Aufgaben und Tätigkeiten, die durch Modelle unterstützt werden können (siehe folgendes Kapitel, Abschnitt 3.1). Umgestaltungsprojekte können nur eine Teilmenge dieser Tätigkeiten, aber auch zusätzliche Tätigkeiten erfordern. Beispielsweise sind Mitarbeiterschulungen in der Praxis von großer Bedeutung.

Das Ziel dieser Arbeit – besser an Umgestaltungsprojekte angepasste Modellierungsansätze zu ermöglichen – kann nicht völlig losgelöst von der Aufbau- und der Ablauforganisation solcher Projekte erreicht werden. Schließlich müssen Modellerstellung und Modellnutzung stimmig in den Gesamtprozess der Umgestaltung eingebettet sein.

Im Rahmen der vom Autor durchgeführten empirischen Untersuchung von Umgestaltungsprojekten wurden mehrere Rollen identifiziert, die typisch für diese Art von Projekt sind:

Projektleitung: Üblicherweise gibt es in einer Projektorganisation mit dem *Projektleiter* eine dedizierte Leitungsrolle für das gesamte Projekt und mehrere *Teilprojektleiter*, die dem Projektleiter unterstellt sind. Diese Rollen werden als *Projektleitung* zusammengefasst.

Architekt: Ein Architekt „entwirft nicht nur die Struktur des zu Schaffenden, sondern leitet und überwacht auch die Realisierung“ [LL13, S.74]. Je nach Art des „zu Schaffenden“ können verschiedene Typen von Architekten unterschieden werden, z.B. der Software-Architekt (für Softwaresysteme) und der Unternehmensarchitekt (für Unternehmensarchitekturen).

Modellierer: Modellierer erfassen Informationen und verarbeiten sie zu Modellen, welche von den anderen Rollen genutzt werden.

Software-Entwickler: In der Regel müssen in Umgestaltungsprojekten bestehende Softwaresysteme angepasst und manchmal auch neue Softwaresysteme entwickelt werden. Beide Aufgaben werden durch *Software-Entwickler* ausgeführt.

Tester: *Tester* minimieren das Risiko der Umgestaltung, indem sie Testfälle konzipieren und durchführen.

Daneben gibt es in der Regel noch weitere projektbezogene Rollen. Aus der Liniorganisation müssen mindestens die Leitung der IT- und Fachabteilungen, die Anwender und die für die Softwaresysteme zuständigen IT-Mitarbeiter ins Projekt einbezogen werden.

Jedes Projekt läuft nach einem bestimmten Vorgehen ab. Die konkrete Organisation eines Projektablaufs hängt unter anderem von dessen Inhalt, von äußeren Rahmenbedingungen und vom durchführenden Unternehmen ab. Eine für Umgestaltungsprojekte geeignete Ablauforganisation kann einen wichtigen Beitrag zur erfolgreichen Umgestaltung einer Anwendungslandschaft leisten. Daher kann nicht einfach eine beliebige Ablauforganisation gewählt werden. Um die in dieser Arbeit angestrebten Bausteine für Modellierungsansätze zu entwickeln, ist es jedoch nicht notwendig, ein eigenes Vorgehen zu konzipieren. Stattdessen wird in diesem Abschnitt ein Typ von Ablauforganisation – ein sogenanntes *Vorgehensmodell* (vgl. [FZ02, S.776ff]) – für Umgestaltungsprojekte festgelegt.

Wie in Kapitel 3 noch gezeigt wird, werden Anwendungslandschaften und ihre Modelle umgestaltet, indem sie durch ihren Gebrauch rückgekoppelt, verändert, erneut rückgekoppelt usw. werden. Das heißt, dass Umgestaltung ein iterativer Prozess ist. Ein für Umgestaltungsprojekte geeignetes Vorgehensmodell muss eine wiederholte Rückkopplung ermöglichen. Solche Vorgehensmodelle werden *zyklische Vorgehensmodelle* (siehe [FZ02]) genannt. Herstellung und Einsatz von Arbeitsergebnissen wie z.B. Modellen sind in diesem Vorgehensmodell verschränkt. Ein

Arbeitsergebnis wird „nicht mehr als ein Produkt, sondern als eine Folge von Versionen verstanden“[FZ02, S.779]. Ein zyklisch organisierter Projektablauf schafft daher die Voraussetzung für gut ins Projekt integrierte Modellierung. Auf der Basis eines solchen Vorgehensmodells kann ein konkretes Projektvorgehen ausgestaltet werden.

2.4 Zusammenfassung

In dieser Arbeit bezieht sich der Begriff „Anwendungslandschaft“ auf ein Mensch-/Aufgabe/Technik-System mit mehreren, zumindest teilweise voneinander abhängigen Softwaresystemen. Ein Teil der Softwaresysteme wird direkt von menschlichen Akteuren genutzt, um ihre Aufgaben im Rahmen betrieblicher Geschäftsprozesse zu erledigen.

Unter der Umgestaltung von Anwendungslandschaften wird eine gezielte Veränderung mehrerer Softwaresysteme verstanden, die sich auf die von der Anwendungslandschaft unterstützten Geschäftsprozesse auswirkt. Solche Vorhaben können unter anderem aus fachlichen, technischen oder gesetzlichen Gründen notwendig werden und stellen in der Regel ein großes Risiko für die durchführende Organisation dar.

Umgestaltung wird in Form von „Umgestaltungsprojekten“ durchgeführt. Dieser Typ von Projekt stellt den Kontext dar, in dem im Rest dieser Arbeit die Modellierung von Anwendungslandschaften betrachtet wird. Damit beantwortet dieses Kapitel die erste Forschungsfragen (*Was charakterisiert Umgestaltungsprojekte?*).

In Umgestaltungsprojekten gibt es eine Reihe typischer Tätigkeiten: Unter anderem müssen Informationen über die Anwendungslandschaft erfasst, das Projekt geplant und die Veränderungen umgesetzt werden. Außerdem müssen Anwendungslandschaften bewertet der konkreten Handlungsbedarf an Softwaresystemen und Geschäftsprozessen festgestellt werden.

Viele dieser Tätigkeiten werden mit Hilfe von Modellen erledigt. Modelle zur Umgestaltung einzusetzen ist jedoch mit einigen Problemen verbunden. Dies zeigt das nächste Kapitel, in dem die Modellierung in Umgestaltungsprojekten analysiert wird.

Kapitel 3

Modellierung in Umgestaltungsprojekten

Ziel dieses Kapitels ist, bewertbar zu machen, wie gut sich existierende Modellierungsansätze für Umgestaltungsprojekte eignen. Dazu wird erläutert, welchem Zweck die Modellierung in Umgestaltungsprojekten dient und die Modellierungsprobleme in solchen Projekten werden empirisch untersucht. Die Ursachen der Modellierungsprobleme werden analysiert und auf vier grundlegende Eigenschaften zurückgeführt, die sich aus dem Gegenstandsbereich und den Rahmenbedingungen der Modellierung ergeben. Abschließend werden aus diesen Grundlagen 13 Anforderungen an Modellierungsansätze abgeleitet, die in Kapitel 4 zur Bewertung existierender Ansätze herangezogen werden.

3.1 Modelle von Anwendungslandschaften

Umgestaltungsprojekte durch Modellierung zu unterstützen ist naheliegend, weil Modelle und Modellierung in einem ingenieurmäßigen Ansatz naturgemäß eine große Rolle spielen (vgl. [Win10, S.4]). Alle Modelle haben drei grundlegende Eigenschaften, die Stachowiak in seiner *Allgemeinen Modelltheorie* folgendermaßen beschreibt:

- *Abbildungsmerkmal*: „Modelle sind stets Modelle von etwas, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können.“ [Sta73, S.131].

- *Verkürzungsmerkmal*: „Modelle erfassen im allgemeinen nicht alle Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder Modellbenutzern relevant scheinen.“ [Sta73, S.132].
- *Pragmatisches Merkmal*: „Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion a) für bestimmte [...] Subjekte, b) innerhalb bestimmter Zeitintervalle und c) unter Einschränkung auf bestimmte gedankliche oder tatsächliche Operationen.“ [Sta73, S.132f].

Den Begriff „Modell“ über diese drei Merkmale zu definieren, ist verbreitet und für beliebige Gegenstandsbereiche – auch Anwendungslandschaften – anwendbar. Betrachtet man Modelle speziell im Gebiet der Informatik, können laut Floyd und Klischewski drei Arten von Informatik-Modellen unterschieden werden (vgl. [FK98, S.2]):

- *Formale Modelle* sind *Spezifikationen*.
- *Berechnungsmodelle* sind *Programme*.
- *Anwendungsmodelle* dienen dazu, einen Gegenstandsbereich zu verstehen und über ihn zu kommunizieren.

Anwendungsmodelle können bei der Umgestaltung vielfältig eingesetzt werden. Andere Autoren bezeichnen diese Art von Modellen auch als *konzeptuelle Modelle* (vgl. [HM08] und [Tha12]) und *konzeptionelle Modelle* (vgl. [BP07]). Im Folgenden wird der Begriff *konzeptuell* verwendet:

Konzeptuelles Modell (Definition 3.1.1): Eine *konzeptuelles* (d.h. ein Konzept aufweisendes) Modell bildet einen Weltausschnitt ab (vgl. [HM08, S.379]). Konzeptuelle Modellierung „ist ein sozialer Prozess, der durch die Intentionen und Anschauungen der beteiligten Personen geprägt ist“ [BP07, S.7f].

Eine besondere Rolle kommt konzeptuellen Modellen bei der Analyse von Anwendungslandschaften zu. Dabei können beispielsweise folgende Fragen relevant sein:

- Wie unterstützt die Anwendungslandschaft die Akteure dabei, ihre Aufgaben im Rahmen eines Geschäftsprozesses umzusetzen?

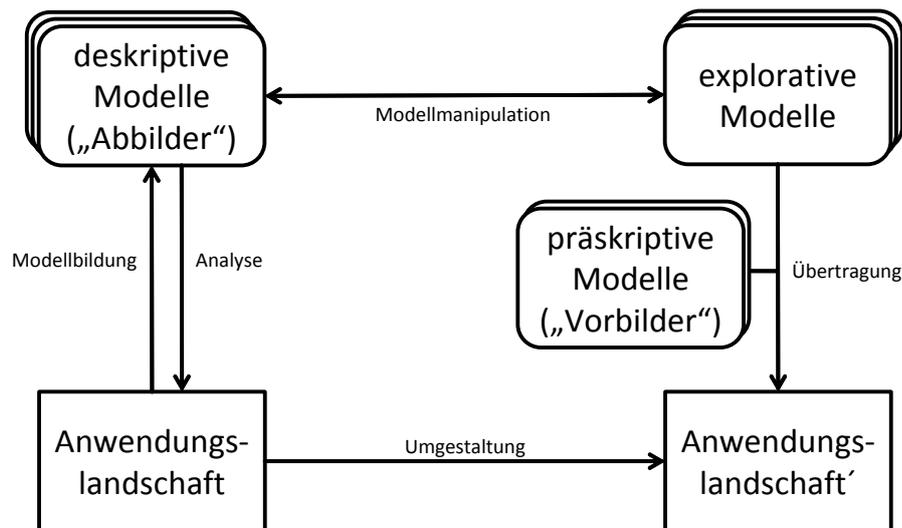


Abbildung 3.1: Umgestaltung einer Anwendungslandschaft mit Hilfe von Modellen (angelehnt an [LL13, S.8])

- Welche IT-gestützten Arbeitsgegenstände werden von Softwaresystemen bereitgestellt, die außer Betrieb genommen werden?
- Stehen diese Arbeitsgegenstände in vergleichbarer Form in den neuen Softwaresystemen zur Verfügung, welche die bisherigen Softwaresysteme ersetzen?
- Welche Softwaresysteme sind indirekt von den Änderungen betroffen?

Konzeptuelle Modelle eignen sich für diese Zwecke, weil sie es ermöglichen, den gegenwärtigen und zukünftigen Einsatz von Softwaresystemen im abgebildeten Weltausschnitt zu analysieren. Wird ein Modell eingesetzt, um zukünftige Zustände von Anwendungslandschaften zu entwerfen, nennt man es auch *exploratives Modell*:

Exploratives Modell (Definition 3.1.2): Eine *exploratives* Modell wird eingesetzt, um „die Folgen einer vorgeschlagenen [...] Änderung der Realität [zu] beurteilen“ [LL13, S.7].

Explorative Modelle sind für Umgestaltungsprojekte von großer Bedeutung. Sie helfen dabei, die geplante Veränderung einer Anwendungslandschaft zu bewerten. Abbildung 3.1 verdeutlicht den Zusammenhang zwischen explorativen Modellen und der modellierten Anwendungslandschaft:

Fragen zum aktuellen Zustand der Anwendungslandschaft können durch *Analyse* von *deskriptiven Modellen* beantwortet werden. Fragen zu möglichen zukünftigen

Zuständen der Anwendungslandschaft lassen sich beantworten, in dem man die deskriptiven Modelle *manipuliert*. Dadurch entstehen explorative Modelle. Verschiedene Manipulationen eines deskriptiven Modells können mehrere explorative Modelle verschiedener Soll-Zustände ergeben. Explorative Modelle ermöglichen *Probehandlungen*, welche die reale Umgestaltung vorwegnehmen. Dies ist nützlich, weil Modellmanipulationen umkehrbar sind. Im Gegensatz dazu kann eine tatsächliche Umgestaltung nicht mehr vollständig zurückgenommen werden. Denn selbst wenn der Ausgangszustand der Anwendungslandschaft wiederhergestellt wird, wurde das Unternehmen durch das gescheiterte Umgestaltungsprojekt und die verbrauchten Ressourcen verändert. Um dies zu verhindern, werden die Auswirkungen von Umgestaltungsprojekten durch Modellmanipulation und Probehandeln an explorativen Modellen abgeschätzt. Werden explorative Modelle auf eine Anwendungslandschaft *übertragen*, ändert sich ihr Zweck. Als *präskriptive Modelle* dienen sie im Sinne eines Bauplans als *Vorbild* für den angestrebten Zustand der Anwendungslandschaft.

Daher sind Modelle insbesondere für jene Umgestaltungsaufgaben relevant, die in Abschnitt 2.3 unter „Bewerten und Entscheiden“ zusammengefasst wurden. Modelle helfen zu entscheiden, welche Umgestaltungsmaßnahmen durchgeführt werden müssen – auf technischer, organisatorischer und fachlicher Ebene.

In der Literatur über Modellierung von Anwendungslandschaft ist es üblich, zwischen dem Modell selbst und seiner grafischen Darstellung zu unterscheiden (z.B. in [BELM08], [Lan13] und [ANS00]). Stachowiak teilt grafische Modelle in [Sta73] in verschiedene Kategorien, wie z.B. *Flussdiagramme*, ein (siehe [Sta73, S.169]). Beim Modell selbst kann es sich zum Beispiel um eine Datenbank handeln, in der Informationen zu einer Anwendungslandschaft strukturiert erfasst werden. Ein solches Modell wird *Informationsmodell* genannt:

Informationsmodell (Definition 3.1.3): Ein *Informationsmodell* enthält Informationen über ein Original. Ein Informationsmodell kann zum Beispiel die Form eines Entity-Relationship-Modells (vgl. [Che76]) in einer Datenbanken annehmen. Im Informationsmodell enthaltene Daten können durch ein oder mehrere grafische Modelle dargestellt werden.

Die Verbindung zwischen einem Informationsmodell und seiner Darstellung in Form grafischer Modelle wird über *Viewpoints* erreicht:

Viewpoint (Definition 3.1.4): Ein *Viewpoint* beschreibt, wie die Daten eines Informationsmodells in grafischen Modellen dargestellt werden (vgl. [BELM08, S.21]).

Sofern nicht explizit anders angegeben, werden die Begriffe *Modell* und *modellieren* in dieser Arbeit im Sinne konzeptueller, grafischer Modelle und konzeptueller, grafischer Modellierung verwendet.

3.2 Modellierungsansätze

Nicht nur fertige Modelle, sondern auch der Vorgang des Modellierens kann helfen, eine Anwendungslandschaft zu verstehen und über bevorstehende Änderungen zu kommunizieren. Dieser Vorgang – das *Modellierungsvorgehen* – und die *Modellierungssprache* werden zusammen oft als *Modellierungsmethode* bezeichnet (vgl. [FGKW07]).

Die *Modellierungssprache* wird laut Fill et al. „durch eine bestimmte Syntax, Semantik und Notation“ [FGKW07, S.421] definiert. Die Semantik legt die Bedeutung der Elemente der Modellierungssprache fest und die Notation die Darstellungsform. Die Syntax kann beispielsweise in Form eines Metamodells beschrieben werden (vgl. [FGKW07]).

Die Analyse von Modellen einer Anwendungslandschaft kann aufwändig sein und oft erst durch Werkzeuge wirtschaftlich lohnenswert werden. Modellierungswerkzeuge werden oft als wichtiger Bestandteil von Modellierungsansätzen genannt. So zählen Karagiannis und Kühn „Mechanismen und Algorithmen“ für eine Werkzeugunterstützung als Bestandteil einer Modellierungsmethode auf (vgl. [KK02, S.185]). Im von Winter beschriebenen Metamodell einer Modellierungsmethode sind Werkzeuge ebenfalls enthalten (vgl. [Win10, S.56]). Werkzeuge sind nicht nur Hilfsmittel für die Erstellung grafischer Modelle und ihre Analyse. In der Linguistik gehört zu jeder Sprachverwendung eine *Pragmatik*, die sich mit der Verwendung der Sprache befasst. Modellierungswerkzeuge prägen die Pragmatik von Modellierungssprachen und beeinflussen so, wie Modellierer eine Modellierungssprache wahrnehmen und mit Modellen *umgehen*. Unter dem Umgang mit Modellen wird in dieser Arbeit Folgendes verstanden:

Umgang mit einem Modell (Definition 3.2.1): Als *Umgang* mit einem Modell wird seine Veränderung (Modellmanipulation) und die manuelle oder werkzeuggestützte Analyse des Modells durch einen Menschen bezeichnet.

In der Praxis trifft man auf informell und unvollständig beschriebene Modellierungssprachen und Vorgehensweisen, sowie auf Modellierungswerkzeuge ohne

explizit definierte Modellierungssprache. All diese können trotzdem nutzbringend in der Praxis eingesetzt werden. In dieser Arbeit werden solche Sprachen, Vorgehensweisen als auch formal beschriebene Modellierungsmethoden unter dem Begriff *Modellierungsansatz* zusammengefasst. Dieser wurde bisher umgangssprachlich gebraucht und wird nun folgendermaßen definiert:

Modellierungsansatz (Definition 3.2.2): Ein *Modellierungsansatz* umfasst

- eine Modellierungssprache,
- ein darauf abgestimmtes Vorgehen für die Modellierung
- ein Vorgehen für den Umgang mit den erstellten Modellen

und wird durch ein Modellierungswerkzeug unterstützt. Diese Bestandteile müssen weder vollständig vorhanden noch explizit beschrieben sein.

3.3 Modellierungsprobleme

Dieser Arbeit liegt die These zu Grunde, dass existierende Modellierungsansätze den Zweck der Modellierung in Umgestaltungsprojekten nicht ausreichend erfüllen. In Abschnitt 1.1 wurde daher die Frage nach Modellierungsproblemen in Umgestaltungsprojekten gestellt. Diese Frage wurde durch eine empirische Untersuchung beantwortet, deren Ergebnisse in diesem Abschnitt beschrieben werden. Ausgangspunkt der Untersuchung waren mehrere typische Umgestaltungsszenarien, in denen Experten mögliche Probleme identifizierten (siehe Anhang A.1). Jene Probleme, die Modelle und Modellierung betreffen, wurden zu zwölf Modellierungsproblemen zusammengefasst und mittels eines Fragebogens durch sieben¹ Experten evaluiert, die in unterschiedlichen Rollen (Projektleiter, IT-Abteilungsleiter, Berater und Software Architekt) an Umgestaltungsprojekten beteiligt waren. Es gab keine Überschneidungen zwischen den beiden Expertengruppen. Die Relevanz der zwölf Probleme wurde mit einer fünfstufigen Skala beurteilt. Die acht Probleme, die im arithmetischen Mittel über dem Mittelwert der Skala lagen, wurden als relevant gewertet. Zwei Probleme wurden im Nachhinein gestrichen, weil sie allgemeine Qualitätskriterien von Modellen betreffen. Die verbliebenen sechs Modellierungsprobleme werden im

¹Die Evaluation wurde nach sieben Teilnehmern beendet, weil sich die Ergebnisse stabilisiert hatten und durch weitere Teilnehmer keine relevanten Abweichungen mehr zu erwarten waren.

Folgenden vorgestellt. Der Fragebogen und die Auswertungsergebnisse finden sich im Anhang A.2.

Modellierungsproblem 1. Modelle müssen darstellen können, *wie* die Anwendungslandschaft die Geschäftsprozesse unterstützt. Es reicht nicht aus, zu zeigen, *dass* ein Softwaresystem einen Prozess unterstützt.

Anwendungslandschaften hängen eng mit den von ihnen unterstützten Geschäftsprozessen zusammen. Ohne adäquate Modelle ist für die Akteure oft nicht nachvollziehbar, wie Softwaresysteme als Werkzeuge eingesetzt werden und welche digitalen Arbeitsgegenstände und Arbeitsergebnisse sie bereitstellen. Modelle werden benötigt, um die Auswirkung von technischen Umstellungen auf Arbeitsabläufe abzuschätzen. Sie ermöglichen außerdem, anwendungsübergreifende Testfälle zu entwickeln. Diese sind notwendig, weil von erfolgreich einzeln getesteten Softwaresystemen nicht darauf geschlossen werden kann, dass komplexe, anwendungsübergreifende Geschäftsprozesse funktionieren.

Modellierungsproblem 2. Die zu modellierende Informationsmenge ist schlecht handhabbar.

Die große Menge an potentiell für ein Umgestaltungsprojekt relevanten Informationen über eine Anwendungslandschaft bewirkt, dass mit vertretbarem Aufwand und in sinnvoller Zeit kein vollständiges Bild einer Anwendungslandschaft zu erheben ist. Selbst die Beschränkung auf Teilbereiche einer Anwendungslandschaft kann dieses Problem nicht immer lösen. Das bedeutet, dass Modellierer und Modellnutzer eine methodische Unterstützung beim Modellieren und im Umgang mit Modellen benötigen.

Modellierungsproblem 3. Bei der Modellierung können widersprüchliche Informationen auftreten.

Modellierer stehen vor dem Problem, dass an der Modellierung beteiligte Personen Aussagen auf Basis von unbegründeten Annahmen treffen und manche Beteiligten ihr Wissen verfälscht weitergeben. Widersprüche sind daher möglich. Beispielsweise verstehen die beteiligten Personen unter einem konkreten Softwaresystem nicht unbedingt dasselbe, nur weil sie dafür den gleichen Namen verwenden.

Modellierungsproblem 4. Abhängigkeiten und Einsatz von Softwaresystemen sind besonders schwierig zu modellieren, wenn die Softwaresysteme abteilungs- und anwendungsübergreifende Geschäftsprozesse unterstützen.

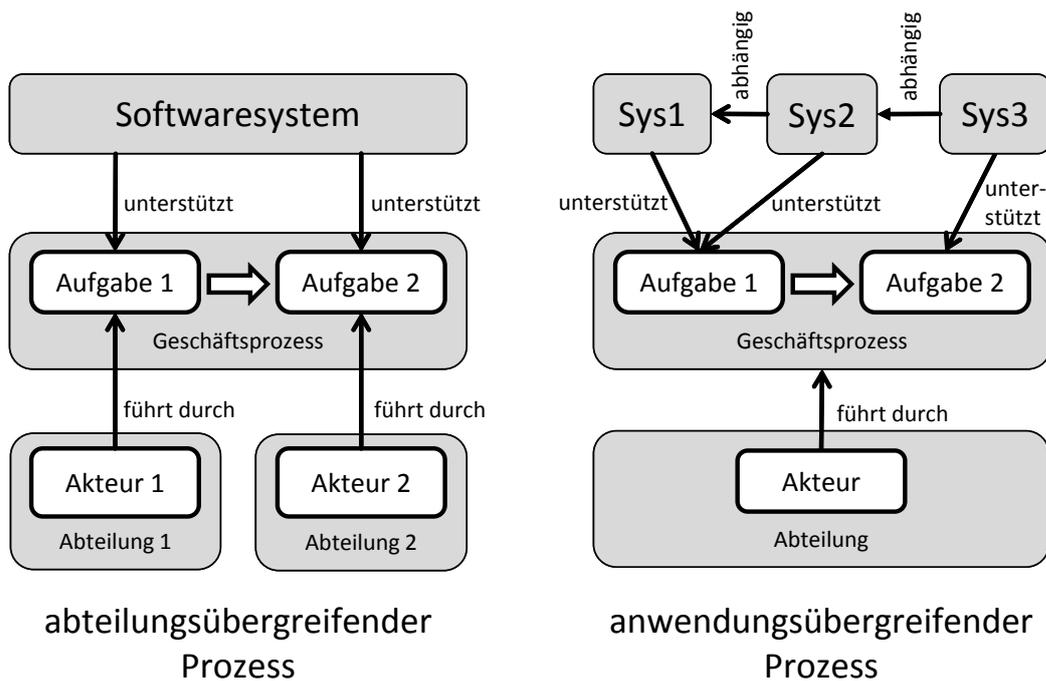


Abbildung 3.2: Abteilungsübergreifender und anwendungsübergreifender Prozess

Abteilungsübergreifende Prozesse sind eine Sonderform kooperativer Arbeit (siehe Abschnitt 2.1). Akteure aus mehreren Organisationseinheiten führen Aufgaben aus, die einen zusammenhängenden Geschäftsprozess ergeben. Zur Kooperationsunterstützung verwenden sie ein Softwaresystem. Anwendungsübergreifende Prozesse bestehen ebenfalls aus mehreren Aufgaben, für die jedoch unterschiedliche Softwaresysteme herangezogen werden. In abteilungs- und anwendungsübergreifenden Geschäftsprozessen arbeiten Akteure über Abteilungsgrenzen hinweg kooperativ und verwenden dazu mehrere Softwaresysteme. Abbildung 3.2 soll diese Zusammenhänge verdeutlichen.

Modellierungsproblem 5. Abhängigkeiten zwischen Softwaresystemen sind schwer abzubilden, wenn sie sich nicht aus technischen Schnittstellen ergeben.

Ein besonderes Problem für die Modellierung ist, Abhängigkeiten zu entdecken, die keine technische Schnittstelle aufweisen (z.B. „Kopplung über die Zeit“, siehe Abschnitt 2.1). In diesen Fällen können Informationen über die Anwendungslandschaft nicht durch technische Hilfsmittel wie der Auswertung von Schnittstellenzugriffen erhoben werden. Daher ist der Modellierer darauf angewiesen, den Einsatz der Anwendungslandschaft zu analysieren und daraus Abhängigkeiten abzuleiten.

Modellierungsproblem 6. Veränderungen einer Anwendungslandschaft über die Zeit müssen dargestellt werden.

Veränderungen müssen so kommuniziert werden, dass die davon betroffenen Akteure wissen, was sich wann in ihren Arbeitsabläufen ändert. Werden Änderungen in mehreren Etappen umgesetzt, müssen diese verschiedenen Zustände dargestellt werden. Je nach Beteiligtem muss diese Information nur fachlich oder auch technologisch sein, beispielsweise um die Auswirkung verschiedener Arten von Integration zwischen Softwaresystemen diskutieren zu können.

3.4 Analyse der Modellierungsprobleme

Die folgenden Abschnitte zeigen, dass die Ursachen der Modellierungsprobleme im Gegenstandsbereich der Modellierung zu verorten sind. Aus der Theorie zu Modellierung und durch Analogien zur Softwaretechnik werden vier Eigenschaften herausgearbeitet, welche die Modellierung in Umgestaltungsprojekten prägen.

3.4.1 Modellierung als Realitätskonstruktion

In Modellen von Anwendungslandschaften und ihres Einsatzes können Widersprüche auftreten (Modellierungsproblem 3). Um herauszufinden, ob dieses Problem verhinderbar ist, muss analysiert werden, aus welchen Quellen die zu modellierenden Informationen stammen:

In Umgestaltungsprojekten erstellte Modelle basieren zumindest teilweise auf Aussagen der Personen, welche die Softwaresysteme verwenden, warten oder entwickeln. Es wäre naiv, anzunehmen, dass diese Personen einem Modellierer alle für ihn relevanten Informationen mitteilen und alle Informationen widerspruchsfrei sind – sonst würde Modellierungsproblem 3 in der Praxis nicht auftreten. Außerdem müssen Modellierer davon ausgehen, dass es relevante Informationen gibt, von deren Existenz sie nicht wissen. Daher ist unsicher, wie aussagekräftig Modelle unter diesen Gegebenheiten überhaupt sein können. Herrmann thematisiert in [HHL99] diese Unsicherheit, die konzeptuellen Modellen inhärent ist: „Der Modellierer hat also Wissen bzgl. der Unsicherheit des zu modellierenden Sachverhaltes, z.B. weil bei der Recherche bestimmte Fragen unbeantwortet bleiben oder weil es zu Widersprüchen kommt.“ [HHL99, S.72]. Konzeptuelle Modelle sind laut Herrmann *vage* (vgl. [HHL99]).

Um die Abhängigkeit von menschlichen Informationsquellen zu verringern, können Anwendungslandschaften und ihr Einsatz vermessen werden. Wie im Folgenden gezeigt wird, stellen jedoch auch automatisiert gesammelte Daten keine objektive, „allwissende“ Instanz dar, die menschliche Informationsquellen ersetzen kann. Zwei Beispiele für Methoden zur Vermessung sind *Process Mining* und *Netzwerk-Scans*:

- Process Mining (vgl. [Aal11]) rekonstruiert Geschäftsprozesse aus Log-Einträgen von Softwaresystemen und wird in der Regel auf Standardsoftware mit ausgeprägter Ablauf-Unterstützung (z.B. ERP-Systeme) angewendet.
- Netzwerkskans werden z.B. im Rahmen von Sicherheits-Audits durchgeführt und können die Port-zu-Port-Verbindungen zwischen Softwaresystemen zeigen. Sie liefern damit Informationen über Abhängigkeiten in einer Anwendungslandschaft.

Automatisiert gesammelte Information sind erst verwertbar, nachdem sie in ein Informationsmodell (z.B. in Form einer CMDB, vgl. Abschnitt 1.2) integriert wurden. Dieses wurde wiederum von einem Modellierer für einen bestimmten Zweck, unter bestimmten Abstraktionen und Annahmen erstellt. Damit sind solche Informationsmodelle konzeptuell (siehe Definition 3.1.1) und daher vage.

Dass konzeptuelle Modelle vage sind, zeigt sich nicht nur in Form von Widersprüchen. In der informatischen Modellierung spielen Vollständigkeit und Korrektheit eine große Rolle bei der Diskussion über die Qualität von Modellen (vgl. [SHP09]). Daraus einen Anspruch nach vollständiger und korrekter Abbildung in Modelle abzuleiten, wäre jedoch falsch, weil konzeptuelle Modellierung „keine exakte Abbildung einer (potentiellen) Realität liefern [kann]“ [BP07, S.7f]. In einer umfangreichen Literaturrecherche haben Mohagheghi et al. verschiedene Ansätze zur Definition von Qualitätsmerkmalen für konzeptuelle Modelle verglichen. Einer der untersuchten Ansätze stellt fest, dass Vollständigkeit und Korrektheit keine realistischen Ziele sind und nur für triviale konzeptuelle Modelle erreicht werden können (vgl. [LSS94, S.46]). Die Autoren dieses Ansatzes, Lindland et al., führen daher den Begriff „feasible“ (im Folgenden als *brauchbar* übersetzt) ein, um Vollständigkeit und Korrektheit zu qualifizieren. Die Idee der Brauchbarkeit von Modellen erlaubt eine Diskussion über Aufwand und Nutzen von Modellierung, auch wenn schwierig zu bewerten ist, wie brauchbar ein Modell ist:

„Feasibility analysis is hard to automate. You’ll always need some gut instinct.“ [LSS94, S.48]

Box und Draper formulieren in [BD87] einen pragmatischen Zugang zu diesem Problem:

„Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.“ [BD87, S.74]

Zusammengefasst können Modelle von Anwendungslandschaften selbst dann brauchbar sein, wenn sie unvollständig, widersprüchlich und fehlerhaft sind.

In diesem Abschnitt wurde eine grundlegende Eigenschaft der Modellierung in Umgestaltungsprojekt identifiziert, die in Anlehnung an Lindland et al.[LSS94] im Folgenden *subjektive Brauchbarkeit* genannt wird:

Eigenschaft 1 (Subjektive Brauchbarkeit). Modelle von Anwendungslandschaften und ihres Einsatzes in Geschäftsprozessen sind vage. Sie stellen keine Fakten dar, sondern eine „konstruierte Realität“. Die Brauchbarkeit der Modelle kann nicht objektiv, sondern nur von den Modellnutzern in Bezug auf den Modellzweck bestimmt werden.

Damit sind Modelle von Anwendungslandschaften mit sogenannten *eingebetteten Programmen* (vgl. [Leh80]) vergleichbar, für die Floyd und Klischewski feststellen, dass ihr Gegenstandsbereich „ein Ausschnitt aus technischer oder sozialer Wirklichkeit ist (z. B. maschinelle Verfahren, Arbeitsabläufe usw.) und [sie] als Informatik-Systeme in der realen Welt zum Einsatz kommen“ [FK98, S.22]. Eine nicht abgeschlossene, unüberschaubare, reale Welt kann auch durch technische Hilfsmittel nicht exakt in Modelle abgebildet werden. Daher kann die Brauchbarkeit von Modellen nicht erhöht werden, in dem man noch detaillierter, umfangreicher und „formaler“ modelliert. Dies würde zu einer noch schlechter handhabbaren Informationsmenge führen (siehe Modellierungsproblem 2). Die folgenden Abschnitte gehen darauf ein, wie brauchbare Modelle erstellt werden können: Indem Modelle als Gestaltungsgegenstand akzeptiert (Abschnitt 3.4.2) und in einem zyklischen, sozialen Prozess modelliert und genutzt werden (Abschnitt 3.4.3).

3.4.2 Modelle als Gestaltungsgegenstand

Eine Anwendungslandschaft entwickelt sich im Lauf der Umgestaltung weiter (vgl. Modellierungsproblem 6) und in der Regel steigt mit der Zeit das Wissen über die

Anwendungslandschaft. Modelle können daher nicht einfach zu Beginn eines Projekts erstellt und dann zur Umgestaltung eingesetzt werden – sie müssen an den jeweils aktuellen Wissensstand angepasst werden. Bisher wurde das Gestalterische an *Umgestaltung* nur auf die Anwendungslandschaft und die von ihr unterstützten Geschäftsprozesse bezogen. Modelle dienen in dieser Betrachtungsweise lediglich als Mittel zum Zweck. Eine „Kluft“ zwischen Modellierung und Einsatz von Modellen birgt das Risiko, dass die Modelle zum Zeitpunkt ihrer Verwendung weniger brauchbar sind. Floyd und Klischewski beschreiben dies in [FK98] allgemein für konzeptuelle Modelle:

„Unvermeidlich ist ferner, daß zwischen der Modellierungssituation und der Einsatzsituation eine unüberwindbare Kluft verbleibt: Zeit vergeht, der Gegenstandsbereich verändert sich, neue Bedingungen entstehen usw.“ [FK98, S.25]

In dieser Arbeit wurde bereits eine Kategorie von Modellen eingeführt, die Gegenstand eines Gestaltungsprozesses sind: Explorative Modelle werden in einem zyklischen Prozess entworfen, eingesetzt und verbessert (siehe Definition 3.1.2). Der Prozess wiederholt sich, bis die Modelle für ihren Zweck – etwa den Einsatz als „Bauplan“ – ausreichend sind. Modellierung kann daher kein einmaliger Schritt in einem Umgestaltungsprojekt sein. Vielmehr ist die Arbeit mit Modellen von Anwendungslandschaften ein wesentlicher Teil der Umgestaltung, der mit der Ablauforganisation des Projekts verzahnt werden muss (siehe Abschnitt 2.3):

Eigenschaft 2 (Gestaltungsgegenstand). Modelle sind nicht bloß Hilfsmittel für die Umgestaltung. Sie sind genauso Gestaltungsgegenstand von Umgestaltungsprojekten wie die Anwendungslandschaft selbst und müssen in einem zyklischen Prozess entworfen, rückgekoppelt und weiterentwickelt werden.

Ein nicht-zyklischer Gestaltungsprozess birgt das Risiko, dass den Modellierern ein *Modellmonopol* (vgl. [Bra73]) zufällt, das anderen Akteuren die Chance auf eine gleichberechtigte Diskussion nimmt. Modellmonopole verhindern, dass Modelle zum Gegenstand eines Lern- und Kommunikationsprozesses werden, der im folgenden Abschnitt als weitere Eigenschaft identifiziert wird.

3.4.3 Modellierung als Lern- und Kommunikationsprozess

Drei der Modellierungsprobleme können auftreten, wenn die in der Anwendungslandschaft arbeitenden Akteure nicht in die Modellierung einbezogen werden:

- Organisatorischen „Gräben“, wie die in Modellierungsproblem 4 geschilderten abteilungs- und anwendungsübergreifenden Abhängigkeiten, sind ohne die Betroffenen schwer zu überwinden.
- Ebenso erhöht sich ohne die Betroffenen das Risiko, dass Modelle nicht konsistent zueinander sind und sich Widersprüche und Lücken auftun (Modellierungsproblem 3).
- Um nicht-technische Abhängigkeiten (Modellierungsproblem 5) zu ermitteln, müssen die in Geschäftsprozessen handelnden Akteure in die Modellierung einbezogen werden.

Der Blickwinkel der Akteure muss in der Modellierung berücksichtigt werden, weil Anwendungslandschaften MAT-Systemen sind. Dass die Akteure zumindest zur Validierung von Modellen einbezogen werden müssen, legt bereits Eigenschaft 2 dar, die Modellierung als zyklische Folge von Entwurf und Rückkopplung charakterisiert. Rückkopplung allein greift jedoch zu kurz. In [FK98] stellen Floyd und Klischewski fest, dass die „von der Modellnutzung Betroffenen [. . .] häufig nicht an der Modellierung beteiligt gewesen [sind] – die Kluft zwischen Entwicklung und Nutzung von Modellen läßt sich durch Dialog allein nicht überbrücken“ [FK98, S.24]. Daher reicht es nicht, Modelle von Anwendungslandschaft im Dialog mit den Akteuren rückzukoppeln. Floyd und Klischewski fordern, dass „Modellbildung als soziale[r] Prozeß verstanden wird, in dem es gilt, die vielfältigen Einflüsse und Rahmenbedingungen [. . .] zu reflektieren und eine gestaltungsförderliche Kommunikationskultur herauszubilden“ [FK98, S.25]. In einem solchen Prozess wirkt sich die Gestaltung der Modelle bereits auf eine Organisation aus, noch bevor deren Anwendungslandschaft tatsächlich verändert wird: „Modelle [. . .] leiten Handlungen an, indem sie Wahrnehmung, Begründungszusammenhänge usw. kanalisieren“ [FK98, S.24].

Um Modellbildung als gestaltungsförderliche Kommunikationskultur zu etablieren, muss verhindert werden, dass ein Akteur ein Modellmonopol hält und die übrigen, *Modell-schwachen* Akteure (vgl. [Bra73]) nicht gleichberechtigt über Modelle

kommunizieren können. Gibt es ein Modellmonopol, kann verstärkte Kommunikation die Modellierungsprobleme sogar verschlimmern:

„With [an actor] in possession of a model monopoly, open information flows and increased communication may increase the influence gap rather than decrease it.“ [Bra73, S.106]

Zusammengefasst liegen die oben genannten Modellierungsprobleme in Umgestaltungsprojekten darin begründet, dass Modellierung nicht als gemeinschaftlicher Lern- und Kommunikationsprozess verstanden wird:

Eigenschaft 3 (Lern- und Kommunikationsprozess). Die in der Anwendungslandschaft arbeitenden Akteure gestalten Modelle in einem Lern- und Kommunikationsprozess. Dieser Prozess verändert die modellierte Organisation.

3.4.4 Architektursichten

Aus mehreren Modellierungsproblemen können Aussagen über die grafische Modellierung in Umgestaltungsprojekten abgeleitet werden:

- Modelle müssen darstellen können, *wie* eine Anwendungslandschaft Geschäftsprozesse unterstützt (Modellierungsproblem 1).
- Modelle müssen nicht-technische Abhängigkeiten darstellen können (Modellierungsproblem 5).
- Modelle müssen darstellen können, wie sich eine Anwendungslandschaft während der Umgestaltung verändert (Modellierungsproblem 6).

Dazu kommt, dass die an einem Umgestaltungsprojekt beteiligten Personen auf Grund ihrer verschiedenen Aufgaben an unterschiedlichen Aspekten der umzugestaltenden Anwendungslandschaft interessiert sind.

In diesem Abschnitt wird untersucht, welche Konsequenzen diese unterschiedlichen Anforderungen an grafische Modelle haben. Eine vergleichbare Problematik existiert in der Softwaretechnik. Sogenannte *Architektursichten* (siehe z.B. [Kru95]) sind ein softwaretechnisches Konzept, um verschiedene Aspekte eines Softwaresystems mit Hilfe unterschiedlicher Modelltypen grafisch darzustellen. Von jedem Modelltyp kann es mehrere konkrete Sichten auf ein Softwaresystem geben, was Abbildung 3.3 verdeutlichen soll.

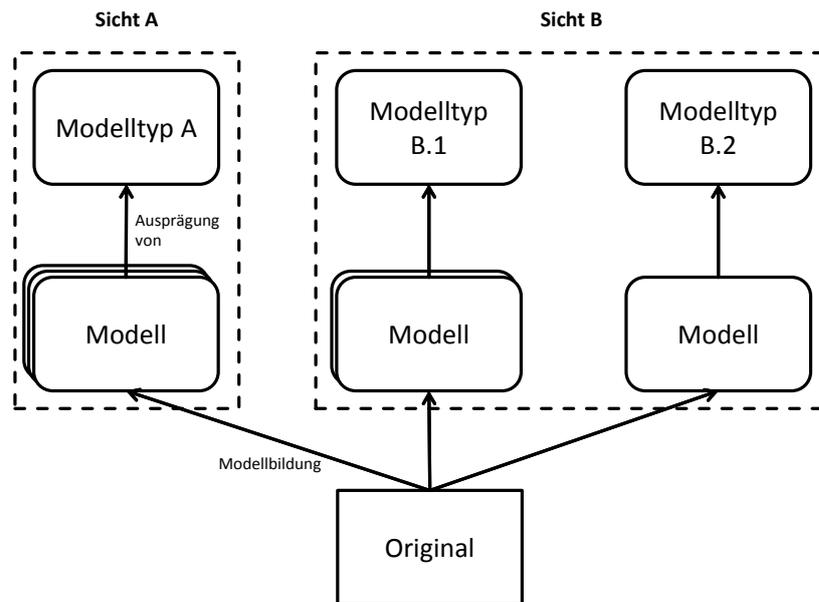


Abbildung 3.3: Modelle als Sichten auf ein Original

Die in der Softwaretechnik verwendeten Sichten werden im Folgenden auf Anwendungslandschaften übertragen. Dabei folgt diese Arbeit der Terminologie² von Gharbi et al. sowie Starke (siehe [GKRS12] und [Sta15]):

- *Bausteinsichten* zeigen die Strukturen und Bausteine der Anwendungslandschaft und damit, wie eine Anwendungslandschaft intern aufgebaut ist.
- *Laufzeitsichten* beschreiben, welche Bausteine zur Laufzeit existieren, wie sie zusammenwirken und damit, wie die Softwaresysteme der Anwendungslandschaft bei der Ausführung von Geschäftsprozessen zusammenarbeiten.
- *Verteilungssichten* „verfolgen zwei Intentionen: 1. Abbildung der technischen Ablaufumgebung des Systems in Form von Hardwarekomponenten (Prozessoren, Speicher, Netzwerke, Router und Firewalls) [und] 2. Abbildung [...] ihrer Softwarebausteine auf die Hardware“ [Sta15, S.172].

Diese drei Arten von Sichten zeigen, wie eine Anwendungslandschaft konstruiert ist und werden im Folgenden als *technologische Sichten* bezeichnet. Da Anwen-

²Diese Terminologie wurde gewählt, weil sie der Ausbildung zum *Certified Professional for Software Architecture* des iSAQB [Int] zugrunde liegt und damit für den deutschsprachigen Raum einen Standard etabliert. Im Englischen ist eine ähnliche Terminologie von Bass et al. [BCK12] verbreitet.

dungslandschaften MAT-Systeme sind (siehe Definition 2.1.5), werden in Umgestaltungsprojekten zusätzliche Sichten benötigt, welche soziale und organisatorische Aspekte beleuchten. Solche Sichten müssen z.B. das Zusammenwirken menschlicher und IT-gestützter Akteure nachvollziehbar machen. Mit der *Kontextabgrenzung* gibt es in Starkes Terminologie eine Sicht, die neben der Architektur eines Systems auch Aspekte wie die Interaktion zwischen System und sogenannten „Stakeholdern“ (z.B. Benutzer) darstellt. Wie Lilienthal in [Lil08] feststellt, ergänzen auch andere Autoren (siehe z.B. [Zül05, S.101ff] und [Kru95]) die technologischen Sichten um weitere Darstellungen, die der „Einbettung des Softwaresystems in den fachlichen Kontext des Anwendungsgebiets“ [Lil08, S.22] dienen. In Anlehnung an Lilienthals Definition werden solche Sichten in dieser Arbeit als *fachliche Sichten* bezeichnet:

fachliche Sichten (Definition 3.4.1): Konzeptuelle Modelle, die den Einsatzkontext einer Anwendungslandschaft – Kernbegriffe, Geschäftsprozesse und die Unterstützung der Anwender – darstellen, werden als *fachliche Sichten* bezeichnet.

Als Bindeglied zwischen fachlichen und technologischen Sichten kann die *fachliche Architektur* aufgefasst werden:

fachliche Architektur (Definition 3.4.2): Eine *fachliche Architektur* ist ein konzeptuelles Modell, da es auf fachlichen Begriffen basiert (vgl. [Zül05, S. 472]) und von technischen Details abstrahiert. Mit Hilfe dieser fachlichen Begriffe wird eine technische Struktur (Bausteinsicht) dargestellt.

Eine fachliche Architektur weist Merkmale von technologischen *und* fachlichen Sichten auf. Sie gibt einen Überblick über einen komplexen Sachverhalt. Im Gegensatz dazu können technologischen Sichten als Konstruktionspläne aufgefasst werden.

Fachliche Sichten und fachliche Architekturen stellen anwendungsorientierte Zusammenhänge dar und zeigen damit, wie eine Anwendungslandschaft Geschäftsprozesse unterstützt – zum Beispiel:

- Welche Arbeitsgegenstände und -ergebnisse werden durch welche Softwaresysteme realisiert?
- Welche Softwaresysteme dienen als Werkzeuge, um Arbeitsgegenstände und -ergebnisse zu bearbeiten?
- Welche Softwaresysteme stellen Informationen bereit, die zur Koordination der Arbeitsabläufe eines Geschäftsprozesses dienen?

Fachliche und technologische Sichten stehen nicht einfach nebeneinander. Sie werden zwar zur Unterstützung verschiedener Aufgaben modelliert, dienen aber einem gemeinsamen, übergeordneten Zweck: Der Umgestaltung einer Anwendungslandschaft. Dieser Zweck verbindet die Sichten miteinander. Verschiedene Sichten auf eine Anwendungslandschaft ergeben nur dann ein konsistentes Gesamtbild, wenn sie für denselben Zweck nützlich sind.

Eigenschaft 4 (Sichten). Anwendungslandschaften können von verschiedenen Standpunkten in Modelle abgebildet werden. Umgestaltungsprojekte erfordern technologische und fachliche Sichten auf eine Anwendungslandschaft. Die verschiedenen Sichten werden durch Modelle ausgedrückt und sind durch das gemeinsame Ziel des Umgestaltungsprojekts miteinander verbunden.

3.5 Anforderungen an Modellierungsansätze

Um zu beurteilen, wie gut sich bestehende Modellierungsansätze für die Modellierung in Umgestaltungsprojekten eignen, werden in diesem Abschnitt Anforderungen an solche Ansätze aufgestellt. Diese Anforderungen leiten sich aus den grundlegenden Eigenschaften der Modellierung in Umgestaltungsprojekten ab, die im vorangegangenen Abschnitt hergeleitet wurden:

- Subjektive Brauchbarkeit von Modellen (Eigenschaft 1)
- Modelle als Gestaltungsgegenstand (Eigenschaft 2)
- Modellierung als Lern- und Kommunikationsprozess (Eigenschaft 3)
- Sichten auf Anwendungslandschaften (Eigenschaft 4)

Als weitere Quelle für Anforderungen an die Sichten dient die in Kapitel 2 eingeführte Begriffswelt, welche eine anwendungsorientierte Perspektive auf Anwendungslandschaften beschreibt (siehe Abbildung 2.5). Eine solche Perspektive muss sich mit den Sichten eines Modellierungsansatzes darstellen lassen. Das heißt, dass aus Abbildung 2.5 Anforderungen an den Sprachumfang von Modellierungsansätzen abgeleitet werden müssen.

Nach welchem Vorgehen die Anforderungen entwickelt wurden, beschreibt Abschnitt 1.3. Folgende Gesichtspunkte wurden bei der Entwicklung berücksichtigt:

- Alle Eigenschaften müssen durch Anforderungen bewertbar gemacht werden.

Anforderung	Subjektive Brauchbarkeit	Gestaltungs- gegenstand	Lern- u. Kommu- nikationsprozess	Sichten
1 IT-gestützte Arbeitsgegenstände				+
2 Manuelle Tätigkeiten				+
3 Koordination				+
4 Fachliche Konzepte			+	+
5 Fachliche Architektur			+	+
6 Technologische Sichten				+
7 Beziehungen zwischen Sichten				+
8 Zweifel an Brauchbarkeit	+			
9 Validierung	+	+	+	
10 Vages Modellieren	+			
11 Einsatzszenarien		+		
12 Modellmanipulation		+	+	
13 Status der Modelle		+		

Tabelle 3.1: Anforderungen als Indikatoren der grundlegenden Eigenschaften

- Alle Bestandteile von Modellierungsansätzen müssen abgedeckt werden: die Modellierungssprache, das Modellierungsvorgehen und das Vorgehen für den Umgang mit Modellen (siehe Definition 3.2.2).
- Jede Anforderung muss nach dem Schema *erfüllt – teilweise erfüllt – nicht erfüllt* bewertbar sein.

Die Werkzeugunterstützung der Modellierungsansätze wird nicht bewertet, weil es für einige Ansätze mehrere Werkzeuge gibt. Die Bewertung der Werkzeugunterstützung wäre in solchen Fällen von der Wahl des Werkzeugs abhängig und daher nur bedingt aussagekräftig.

Tabelle 3.1 zeigt einen Überblick über die Anforderungen, die in den folgenden Unterabschnitten hergeleitet und anhand eines durchgehenden Beispiels erläutert werden:

Die Pfefferminzia Versicherung will die Bearbeitungszeit von Vorgängen wie z.B. der Schadensregulierung verkürzen und zudem die Auskunftsfähigkeit über den Status der Vorgänge erhöhen. Erreicht werden sollen diese Ziele, in dem die Anwendungslandschaft um ein Workflow-System ergänzt wird. Dieses soll die IT-Unterstützung der Vorgangsbearbeitung verbessern.

Grundlage des Beispiels ist ein reales Projekt³, das in der Diplomarbeit von Rost (siehe [Ros09]) beschrieben wird. Es handelt sich dabei um ein Umgestaltungsprojekt, weil in mehrere Softwaresysteme der Anwendungslandschaft eingegriffen und ihre Kopplung verändert wurde. Dadurch wurden die Arbeitsabläufe der betroffenen Akteure stark verändert. Auch Umfang und Art der notwendigen Tätigkeiten entsprechen den Merkmalen eines Umgestaltungsprojekts. Unter anderem mussten Anwendungslandschaft und Geschäftsprozesse der Pfefferminzia erhoben und neu gestaltet werden. An einem Ausschnitt des Geschäftsprozesses „Leistungsfall bearbeiten“ werden der Ist-Zustand der Anwendungslandschaft und ihr angestrebter Soll-Zustand vorgestellt. Diese Zustände zu modellieren stellt eine realistische Aufgabe für einen Modellierer dar, auch wenn der Geschäftsprozess im Vergleich zu [Ros09] vereinfacht wurde und die Informationen nicht erst gewonnen werden müssen, sondern bereits in Textform vorliegen.

In der Abteilung *Posteingang* werden per Post eingegangene, papierhafte Dokumente gescannt und im System *Dokumentverwaltung* digital erfasst. Anhand des Nachnamens der versicherten Person benachrichtigt die Dokumentverwaltung per Mail die für den betroffenen Buchstabenkreis zuständige Gruppe von Sachbearbeitern. Der nächste freie Sachbearbeiter beginnt mit der Bearbeitung des Falls und legt dazu mit *Microsoft Excel* eine Vorgangsmappe für den Fall an. Anhand der Dokumente zum Leistungsfall in der Dokumentverwaltung und den im *Leistungssystem* hinterlegten Leistungsdaten der versicherten Person wird die zu erbringende Leistung geprüft. Das Ergebnis der Prüfung wird in der Vorgangsmappe dokumentiert. Falls die zu erbringenden Leistungen nicht restlos geklärt werden können, gibt der Sachbearbeiter die Vorgangsmappe weiter an einen Leistungsspezialisten. Dieser prüft den unklaren Leistungsfall anhand der Dokumente in der Doku-

³Der Hintergrund des Projekts wird in Abschnitt 6.2.3 noch ausführlicher erläutert. Für das Verständnis des Beispiels ist er aber nicht von Bedeutung.

mentverwaltung. Anschließend dokumentiert er sein Ergebnis in der Vorgangsmappe und gibt diese zurück an den Sachbearbeiter.

Zukünftig sollen die Dokumenten eines Leistungsfalls nicht nur digitalisiert werden, sondern dabei automatisch ein Vorgang im neuen Workflow-System angelegt werden. Das Workflow-System prüft selbstständig die zu erbringende Leistung und dokumentiert das Ergebnis in der Vorgangsmappe. Falls der Leistungsfall nicht restlos geklärt werden kann, informiert das Workflow-System per Mail einen Leistungsspezialisten. Dieser prüft den unklaren Leistungsfall anhand der Dokumente in der Dokumentverwaltung. Anschließend dokumentiert er sein Ergebnis in der Vorgangsmappe und signalisiert dem Workflow-System damit, dass es nun die Bearbeitung fortführen kann.

3.5.1 Anforderungen an die Modellierungssprache

Die Anforderungen an die Modellierungssprache ergeben sich überwiegend aus Eigenschaft 4 (Sichten). Diese Eigenschaft beschreibt, dass fachlichen Sichten und fachliche Architekturen anwendungsorientierte Zusammenhänge einer Anwendungslandschaft darstellen. Um sicherzustellen, dass eine Modellierungssprache anwendungsorientierte Zusammenhänge ausdrücken kann, werden die Anforderungen 1 – 3 aufgestellt:

Anforderung 1. Arbeitsgegenstände und -ergebnisse müssen grafisch mit Softwaresystemen in Beziehung gesetzt werden können.

Im Beispiel: Beim Vergleich von Ist- und Soll-Modelle muss erkennbar sein, dass die digitale Vorgangsmappe im Ist-Zustand mit Excel realisiert wird und im Soll-Zustand mit Hilfe des Workflow-Systems.

Anforderung 2. Manuelle und IT-gestützten Tätigkeiten müssen unterscheidbar sein.

Im Beispiel: Der Umgang mit papierhaften Dokumenten zu Leistungsfällen muss anders dargestellt werden als der Umgang mit digitalisierten Dokumenten.

Anforderung 3. Es muss darstellbar sein, wie Akteure miteinander kommunizieren, um Arbeitsabläufe zu koordinieren.

Im Beispiel: Sachbearbeiter und Leistungsspezialist tauschen eine Vorgangsmappe aus.

Ein Modellierungsansatz für Umgestaltungsprojekte muss nicht nur die anwendungsorientierten Aspekte darstellen, die sich aus dem Einsatz einer Anwendungslandschaft ergeben. Um Modellierung zu einem Lern- und Kommunikationsprozess (Eigenschaft 3) zu machen, muss die Modellierungssprache für ein breites Spektrum an Akteuren verständlich sein. Das setzt weitere fachliche Sichten (Anforderung 4) und fachliche Architekturen (Anforderung 5) voraus, in denen die Akteure sich, ihre Arbeitsabläufe und ihre Arbeitsgegenstände wiedererkennen können. Damit wird das Risiko gesenkt, dass Modellierer ein Modellmonopol halten und die Kommunikation dominieren:

Anforderung 4. Fachliche Konzepte müssen zueinander in Beziehung gesetzt werden können.

Im Beispiel: Die beiden fachlichen Konzepte *Leistung* und *Leistungsfall* müssen einander zugeordnet werden können.

Anforderung 5. Fachliche Architekturen müssen dargestellt werden können.

Im Beispiel: Eine fachliche Architektur der Pfefferminzia Anwendungslandschaft muss die Softwaresysteme *Dokumentverwaltung*, *Leistungssystem* und *Workflow-System*, ihre Beziehungen zueinander und die von ihnen verwalteten Leistungen, Leistungsfälle und Vorgangsmappen zeigen.

Neben fachlichen Sichten werden für eine Umgestaltung auch technologische Sichten benötigt. Die verschiedenen Sichten auf eine Anwendungslandschaft dürfen nicht einfach nebeneinanderstehen (Eigenschaft 4):

Anforderung 6. Technologische Sichten müssen dargestellt werden können.

Im Beispiel: Die Schnittstelle zwischen der Dokumentverwaltung und dem Workflow-System muss abgebildet werden können.

Anforderung 7. Fachliche und technologische Sichten müssen miteinander in Beziehung gesetzt werden können.

Im Beispiel: Aus der Beschreibung des Soll-Prozesses geht hervor, dass das Workflow-System Zugriff auf das Leistungssystem haben soll. Diese fachliche Sicht muss mit einer technologischen Sicht, welche die Schnittstelle zwischen den beiden Systemen beschreibt, verknüpft werden können.

Die Modellierung in Umgestaltungsprojekten ist mit einem gewissen Grad an Unsicherheit verbunden (vgl. Eigenschaft 1), der in Modellen ausdrückbar sein muss:

Anforderung 8. Der Modellierer muss ausdrücken können, an welchen Stellen eines Modells er an dessen Brauchbarkeit zweifelt.

Im Beispiel: Wie sich die Sachbearbeiter untereinander koordinieren, geht nicht aus der Prozessbeschreibung hervor. Ein Modell des Geschäftsprozesses „Leistungsfall bearbeiten“ muss ausdrücken können, dass Informationen fehlen.

3.5.2 Anforderungen an das Modellierungsvorgehen

Mehrere der grundlegenden Eigenschaften erfordern, Modelle während der Modellierung zu validieren. Dadurch ist es möglich, Widersprüche und fehlende Informationen bereits bei der Modellierung zu erkennen (Eigenschaft 1). Modellierung und Validierung zu verzahnen, führt notwendigerweise zu einem Lern- und Kommunikationsprozess der Beteiligten (Eigenschaft 3), in dem Modelle gestaltet werden (Eigenschaft 2):

Anforderung 9. Modelle müssen bereits im Modellierungsprozess validiert werden.

Im Beispiel: Das Modellierungsvorgehen muss ermöglichen, Sachbearbeiter und Leistungsspezialisten am Modellierungsprozess zu beteiligen.

Die nächste Anforderung schließt an Anforderung 8 an. Darin wurde von der Modellierungssprache gefordert, dass sie die subjektive Brauchbarkeit von Modellen ausdrücken können muss (Eigenschaft 1). Damit die geforderten Sprachmittel sinnvoll eingesetzt werden können, muss die subjektive Brauchbarkeit von Modellen auch im Modellierungsvorgehen berücksichtigt werden:

Anforderung 10. Es muss explizit darauf eingegangen werden, dass konzeptuelle Modelle immer vage und nicht zwingend vollständig, korrekt und widerspruchsfrei sind.

Im Beispiel: Die Beschreibung der Abläufe ermöglicht, den Einsatz des neuen Workflow-Systems im Geschäftsprozess „Leistungsfall bearbeiten“ konzeptuell zu modellieren. Der Ansatz muss verdeutlichen, dass diese Modelle nicht geeignet sind, um das Workflow-System zu steuern.

3.5.3 Anforderungen an den Umgang mit Modellen

Modelle werden nicht nur erstellt, sondern auch für bestimmte Zwecke eingesetzt und dadurch rückgekoppelt (siehe Eigenschaft 2 – Gestaltungsgegenstand). Daher

muss ein Modellierungsansatz für Umgestaltungsprojekte den Umgang mit Modellen beschreiben (siehe Definition 3.2.2). Die Anforderungen 11 und 12 gehen auf den Umgang mit Modellen ein: Wie helfen die Modellierungsansätze dabei, für Umgestaltungsprojekte relevanten Tätigkeiten durchzuführen (vgl. Abschnitt 2.3)?

Anforderung 11. Der Ansatz muss konkrete Einsatzszenarien beschreiben, die sich auf Umgestaltungsprojekte anwenden lassen.

Im Beispiel: Die IT der Pfefferminzia muss mit Hilfe des Modellierungsansatzes in der Lage sein, Prozesse zu identifizieren, die einen aussagekräftigen *proof-of-concept* des Workflow-Systems ermöglichen.

Anforderung 12. Der Ansatz muss mögliche Modellmanipulationen und Modellanalysen beschreiben.

Im Beispiel: Ist- und Soll-Modelle müssen miteinander verglichen werden können.

In Umgestaltungsprojekt werden Modelle in der Regel in einem zyklischen Prozess entworfen und rückgekoppelt (Eigenschaft 2). Für einen solchen Prozess ist es hilfreich, Modelle in unterschiedlichen Entwicklungsständen im Werkzeug unterscheiden zu können:

Anforderung 13. Der Status der Modelle im Gestaltungsprozess muss explizit gekennzeichnet werden können.

Im Beispiel: Das Modell des Ist-Prozesses muss als Ist-Modell mit noch zu klärender Unschärfe gekennzeichnet werden können.

3.6 Zusammenfassung

In diesem Kapitel wurde untersucht, welchen Zweck die Modellierung in Umgestaltungsprojekten verfolgt. Außerdem wurde die dritte Forschungsfrage beantwortet – welche Modellierungsprobleme in der Praxis auftreten. An diese Frage schließt die vierte Forschungsfrage (Ursachen der Modellierungsprobleme) an:

Die für diese Arbeit relevanten *konzeptuellen Modelle* bilden einen Gegenstandsbereich ab – Anwendungslandschaften und ihren Einsatz in Geschäftsprozessen. Modelle einer Anwendungslandschaft basieren nicht nur auf messbaren Fakten, sondern auch auf Aussagen der Menschen, welche die Softwaresysteme verwenden oder betreuen. Der Gegenstandsbereich kann daher nur vage modelliert werden (Eigenschaft 1). Modellierung muss im Kontext von Umgestaltungsprojekten als sozialer

Prozess verstanden werden (Eigenschaft 3), in dem die Beteiligten unterschiedliche Sichten auf eine Anwendungslandschaft konstruieren (Eigenschaft 4). Ein solcher Prozess ist notwendigerweise zyklisch (Eigenschaft 2), d.h. durch den Umgang mit den konstruierten Modellen erhalten die an der Modellierung Beteiligten Rückmeldung über die Nützlichkeit der Modelle. So können die Modelle inkrementell verbessert werden.

Ein Modellierungsansatz, der all dies ermöglicht, besteht aus einer Modellierungssprache, einem Modellierungsvorgehen und einer Beschreibung des Umgangs mit Modellen. Für die Praxistauglichkeit des Ansatzes ist zudem eine Werkzeugunterstützung von Bedeutung.

Damit kann zusammengefasst werden, dass die Modellierungsprobleme nur zu lösen sind, wenn die in diesem Kapitel beschriebenen grundlegenden Eigenschaften der Modellierung in Umgestaltungsprojekten berücksichtigt werden. Um dies überprüfbar zu machen, wurde ein Anforderungskatalog entwickelt. Erfüllt ein Ansatz die Anforderungen nicht, wird auch seine umfassendere und stärker formalisierte Anwendung nicht bei der Lösung der Modellierungsprobleme helfen.

Kapitel 4

Bewertung existierender Modellierungsansätze

Ziel dieses Kapitel ist, herauszufinden, wie gut existierende Modellierungsansätze für Umgestaltungsprojekte geeignet sind. Dazu werden relevante Ansätze ausgewählt und anhand des im vorangegangenen Kapitel ausgearbeiteten Anforderungskatalogs bewertet.

4.1 Bewertungsziele und Auswahl der Ansätze

Die Bewertung existierender Modellierungsansätze verfolgt zwei Ziele:

Zum einen wurde in Abschnitt 1.1 die These formuliert, dass existierende Ansätze nicht speziell genug an Umgestaltung angepasst sind. Die Bewertung soll die Defizite der Ansätze aufzeigen und damit den Bedarf an besser angepassten Ansätzen belegen.

Zum anderen soll im konstruktiven Teil dieser Arbeit (siehe Kapitel 5) Bewährtes aus existierenden Ansätzen wiederverwendet und neu kombiniert werden, um Modellierungsansätze besser an Umgestaltungsprojekte anzupassen. Die Bewertung zeigt, welche Vorgehensweisen und Konzepte der Modellierungssprachen die Anforderungen erfüllen und daher für den konstruktiven Teil dieser Arbeit relevant sind.

Wie bereits in Kapitel 1.2 beschrieben wurde, sind Anwendungslandschaften ein Querschnittsthema diverser informatischer und wirtschaftsinformatischer Fachrichtungen. Entsprechend beschäftigen sich viele Ansätze aus Wissenschaft und Praxis – von bloßen Notationen bis hin zu kommerziellen Modellierungswerkzeu-

gen – mit der Darstellung von Anwendungslandschaften. Da ein Vergleich aller vorhandenen Ansätze so gut wie unmöglich ist, werden nur Ansätze zur *Unternehmensmodellierung* (vgl. Abschnitt 1.2) in Betracht gezogen. Solche Ansätze stellen Geschäftsprozesse, Anwendungslandschaften und Organisationsstrukturen dar und haben daher einen stärkeren inhaltlichen Bezug zu Umgestaltungsprojekten als z.B. die aus der Softwaretechnik stammende *Unified Modeling Language (UML)*, siehe [RJB05]).

Ein Ansatz zur Unternehmensmodellierung wurde an der Universität Hamburg von der Gruppe um Heinz Züllighoven entwickelt: Die *exemplarischen Geschäftsprozessmodellierung (eGPM)*. Als Mitglied der Gruppe ist der Autor an der Weiterentwicklung der eGPM beteiligt. Wie in dieser Arbeit spielt das Konzept der Anwendungsorientierung für die eGPM eine bedeutende Rolle. Daher liegt es nahe, die eGPM für die Bewertung in Betracht zu ziehen.

Um weitere Kandidaten für die Bewertung auszuwählen, werden Literaturstudien und Methodenvergleiche anderer Autoren herangezogen. In den Arbeiten von Leist-Galanos [LG06], Braun [Bra07] und Wilfing [Wil13] werden folgende Ansätze verglichen:

- Die *Architektur integrierter Informationssysteme (ARIS)* von Scheer [Sch02].
- Unternehmensmodellierung nach dem *Business Engineering-Ansatz (BEN)* von Winter [Win10].
- Die *multiperspektivische Unternehmensmodellierung (MEMO)* von Frank [Fra12].
- Das *semantische Objektmodell (SOM)* von Ferstl und Sinz [FS93].

Als weitere Quelle für Kandidaten dient eine umfangreiche¹ Literaturstudie von Mykhashchuk et al. (siehe [MBDS11]). Darin werden die fünf meistzitierten Forschungsgruppen aus dem Bereich Unternehmensarchitektur identifiziert. Zwei dieser fünf Gruppen haben Modellierungsansätze entwickelt:

- Die Gruppe um Robert Winter (Universität St. Gallen) mit dem *Business Engineering Navigator* (siehe oben).

¹Die Studie umfasst 299 englischsprachige Veröffentlichungen aus den Jahren 1990 – 2009.

- Die Gruppe um Florian Matthes (Technische Universität München) mit dem *EAM Pattern Catalog* [BELM08].

Die Forschungsgruppe, welche die Modellierungssprache *ArchiMate* (siehe [The13b]) entwickelte, wird in der Literaturstudie erwähnt, schaffte es aber nicht in die fünf meistzitierten Quellen. Das könnte daran liegen, dass die Entwicklung von *ArchiMate* 2008 an *The Open Group* übergeben wurde und die ursprüngliche Forschungsgruppe danach weniger zu *ArchiMate* veröffentlicht hat. Die Studie berücksichtigt keine Veröffentlichungen der Open Group zu *ArchiMate*. Seit 2009 gibt es einen von der Open Group veröffentlichten Sprachstandard für die registriert Marke *ArchiMate*[®]. Als standardisierte Modellierungssprache für Unternehmensarchitektur ist *ArchiMate* ebenfalls ein Kandidat für die Bewertung.

Tabelle 4.1 zeigt einen Überblick der sieben, für die Bewertung in Frage kommenden Ansätze. Da der Aufwand für die Bewertung sehr hoch ist, beschränkt sich diese Arbeit auf die Ansätze mit der größten praktischen Bedeutung. Als Indikator dafür dient, ob kommerzielle Modellierungswerkzeuge verfügbar sind. Bei *ArchiMate*, *ARIS*, *BEN* und *eGPM* ist dies der Fall, weswegen sie in den Vergleich einbezogen werden. Das Werkzeug für *BEN* wird allerdings seit 2010 nicht mehr kommerziell vertrieben und steht kostenfrei im Rahmen der *Open Models Initiative* der Universität Wien [Ope] zur Verfügung. Ebenfalls über die *Open Models Initiative* erhältlich sind Werkzeuge für *SOM* und *MEMO*. Weil diese beiden Werkzeuge aber nur Prototypen sind, werden *SOM* und *MEMO* nicht in den Vergleich einbezogen. Auf den *EAM Pattern Catalog* kann das Kriterium „kommerzielles Modellierungswerkzeug“ nicht angewendet werden, da der Ansatz unternehmerische Praxis in Form von Mustern beschreibt. Die praktische Bedeutung des Ansatzes ist damit gegeben, weswegen er in der Bewertung berücksichtigt wird.

Die Bewertung umfasst somit fünf Ansätze – und damit mehr als die oben zitierten Vergleiche von Leist-Galanos, Braun und Wilfing. Die fünf Ansätze haben ihre Wurzeln in der akademischen Forschung und haben sich in der Praxis bewährt. Deswegen gib es zu diesen Ansätzen wissenschaftliche Publikationen und praxiserprobte Werkzeuge. Auf Grund der vom Autor gewählten Kriterien fanden rein kommerzielle und rein akademische Ansätze keine Berücksichtigung. Damit ist jedoch keine Aussage über ihre Eignung für Umgestaltungsprojekte verbunden. Im Ausblick dieser Arbeit (siehe Kapitel 7) wird weitere Forschung zu diesem Aspekt angeregt.

	erwähnt in	kommerz. Werkzeug
ArchiMate	[MBDS11]	+
ARIS	[LG06] [Bra07] [Wil13]	+
BEN	[LG06] [MBDS11] [Wil13]	(+)
EAM Patterns	[LG06] [MBDS11]	(+)
eGPM	–	+
MEMO	[Bra07] [Wil13]	–
SOM	[Bra07] [Wil13]	–

Tabelle 4.1: Kandidaten für die Bewertung

4.2 Bewertung

Um zu Ergebnissen zu gelangen, wertete der Autor Literaturquellen aus und löste mit jedem Ansatz die gleiche Modellierungsaufgabe. Als Beispiel dient das Umgestaltungsprojekt einer Versicherung, anhand dessen in Abschnitt 3.5 schon die Anforderungen erläutert wurden. Einige der Beispielm Modelle sind im Folgenden zur Veranschaulichung abgebildet. Die detaillierten Bewertungsergebnisse und alle Beispielm Modelle finden sich in Anhang B.

Die Ansätze werden in alphabetischer Reihenfolge vorgestellt und nach folgendem Schema bewertet:

- + Anforderung erfüllt.
- Anforderung teilweise erfüllt.
- Anforderung nicht erfüllt.

Tabelle 4.2 zeigt die Ergebnisse der Bewertung. Zusammengefasst eignen sich eGPM, ArchiMate und die EAM-Patterns am besten für Umgestaltungsprojekte. Aber selbst diese Ansätze erfüllen nur einen Teil der Anforderungen.

4.2.1 ArchiMate

ArchiMate ist eine Modellierungssprache zur Darstellung von IT-Unternehmensarchitekturen, was Anwendungslandschaften mit einschließt. Die Spezifikation von ArchiMate definiert keine Modelltypen. Stattdessen ist die Modellierungssprache in

Anforderung	Archi-Mate	ARIS	BEN	EAM-Patterns	eGPM
1 IT-gestützte Arbeitsgegenstände	●	–	–	●	+
2 Manuelle Tätigkeiten	●	●	●	–	+
3 Koordination	●	–	–	●	+
4 Fachliche Konzepte	+	+	+	●	+
5 Fachliche Architektur	+	–	+	+	+
6 Technologische Sichten	+	–	–	+	–
7 Beziehungen zw. Sichten	+	●	+	●	+
8 Zweifel an Brauchbarkeit	–	–	–	–	+
9 Validierung	–	–	–	–	+
10 Vages Modellieren	–	●	–	–	+
11 Einsatzszenarien	+	●	+	+	●
12 Modellmanipulation	–	●	–	+	●
13 Status der Modelle	–	●	–	–	+

Tabelle 4.2: Vergleich der bewerteten Modellierungsansätze

mehrere Schichten eingeteilt. Im selben Modell können mehrere Schichten visualisiert werden. Die für die Bewertung wesentlichen Schichten und Elementtypen der ArchiMate-Version 2.1 sind:

Business Layer: Business Event, Business Role, Business Process, Business Function, Business Object, Representation

Application Layer: Application Component, Data Object

Relationships: Realization, Access, Association, Used By, Flow, Triggering

Stand September 2014 waren sieben Modellierungswerkzeuge von der Open Group für ArchiMate zertifiziert (vgl. [The14]). Auf Grund einer kostenfreien Evaluierungslizenz wurde aus diesen sieben Werkzeugen BiZZdesign Architect 4.5.2 (vgl. [Biz]) für die Modellierung der Beispiele ausgewählt.

Wie man mit den Modellelementen der ArchiMate-Schichten verschiedene Sichten darstellt, beschreibt die Open Group in Viewpoints (siehe [The13c]). Mehrere dieser Viewpoints eignen sich, um technologische Sichten und zum Teil auch fachliche Sichten auf Anwendungslandschaften zu erstellen. Um weitere fachliche Sichten

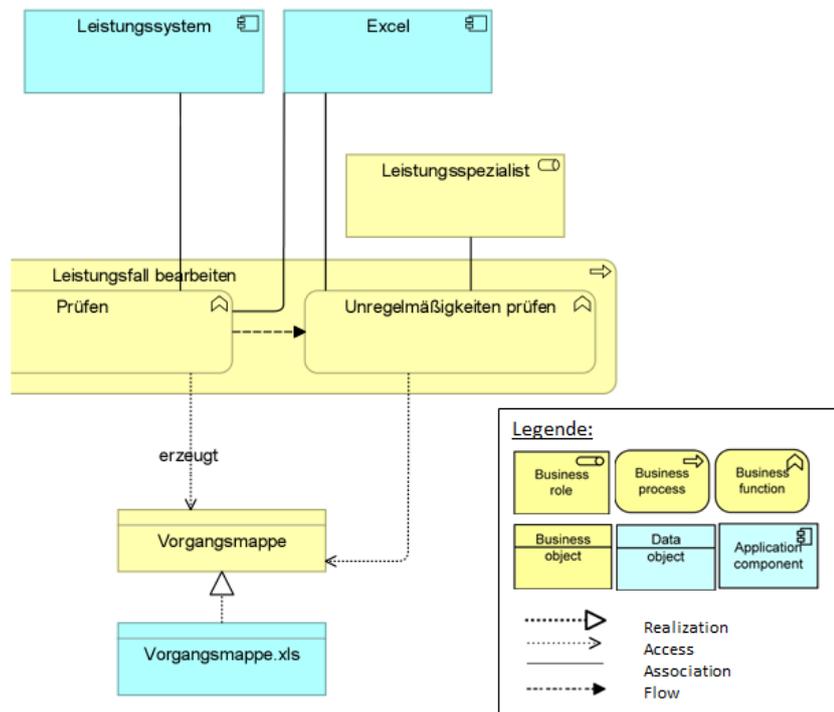


Abbildung 4.1: Ausschnitt eines Geschäftsprozessmodells mit ArchiMate (vollständige Grafik siehe Abbildung B.2)

und fachliche Architekturen modellieren zu können, müssen Viewpoints kombiniert werden. Abbildung 4.1 zeigt eine fachliche Sicht in Form eines Geschäftsprozessmodells, für das drei Viewpoints kombiniert wurden.

ArchiMate eignet sich nur eingeschränkt, um eine anwendungsorientierte Sicht auf Geschäftsprozessmodelle zu modellieren. Zwar gibt es im Business Layer und Application Layer geeignete Elementtypen, doch kann nicht direkt dargestellt werden, welche Arbeitsgegenstände durch Softwaresysteme realisiert und wie Tätigkeiten durch Softwaresysteme unterstützt werden. Das Beispiel in Abbildung 4.1 veranschaulicht, dass der als Business Object modellierte Arbeitsgegenstand „Vorgangsmappe“ nicht direkt mit dem Softwaresystem „Excel“ in Beziehung steht. Dieser Zusammenhang zwischen dem Arbeitsgegenstand und dem Softwaresystem ergibt sich nur indirekt, indem die Vorgangsmappe im Modell durch ein Data Object „Vorgangsmappe.xls“ realisiert wird. Um den Zusammenhang zwischen dem Data Object und der Application Component „Excel“ explizit darzustellen, müsste ein vierter Viewpoint in das Modell integriert werden, der eine fachliche Architektur darstellt (siehe Abbildung 4.2). Doch je mehr Viewpoints in einem Modell über-



Abbildung 4.2: Ausschnitt eines fachlichen Architekturmodells mit ArchiMate (vollständige Grafik siehe Abbildung B.1)

einandergelegt werden, desto unübersichtlicher und komplexer wird es. Wie Excel benutzt wird, um die im Beispiel auftretenden Unregelmäßigkeiten zu prüfen, kann nicht dargestellt werden.

Da ArchiMate lediglich eine Modellierungssprache spezifiziert und kein Modellierungsvorgehen beschreibt, können die Anforderungen mit Bezug zum Vorgehen nicht erfüllt werden: Welchen Status im Gestaltungsprozess ein Modell hat und wie es validiert werden kann, thematisiert ArchiMate nicht.

Der Umgang mit Modellen wird in [The13c] beschrieben, in dem die Einsatzmöglichkeiten der Viewpoints dargelegt und anhand von Beispielen erläutert werden. Auf die Brauchbarkeit von Modellen geht ArchiMate nicht ein.

4.2.2 ARIS

Der Ansatz der *Architektur integrierter Informationssysteme* wurde entwickelt, um Geschäftsprozesse und ihre Unterstützung durch Softwaresysteme im Unternehmenskontext zu betrachten (vgl. [Sch02]). Dazu sind in ARIS fünf Sichten auf Unternehmen definiert:

- Die *Funktionssicht* umfasst die Ziele eines Unternehmens und notwendige Tätigkeiten, um die Ziele zu erreichen.
- Die *Organisationssicht* betrachtet die Aufbauorganisation.
- In der *Datensicht* werden die im Unternehmen verarbeiteten Informationen zusammengefasst.
- Die *Leistungssicht* umfasst unter anderem Produkte und Dienstleistungen.
- Die *Steuerungssicht* (auch: *Prozesssicht*) zeigt, wie die zuvor genannten Aspekte eines Unternehmens durch Geschäftsprozesse zusammenhängen. Sie „bildet

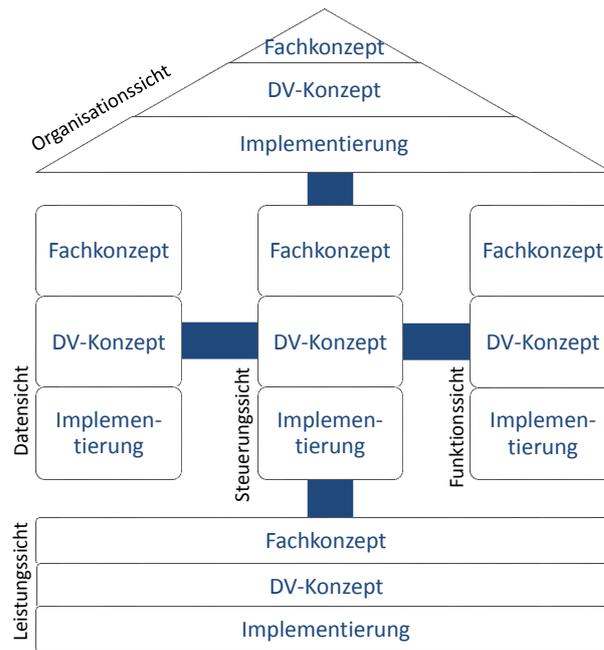


Abbildung 4.3: ARIS-Haus, angelehnt an [Sch02, S.41]

den Rahmen für die [...] Betrachtung aller bilateralen Beziehungen der Sichten sowie der vollständigen Prozessbeschreibung“ [Sch02].

Jede der Sichten kann auf drei Ebenen ausgestaltet werden:

- Auf Ebene eines *Fachkonzepts*,
- auf Ebene einer Spezifikation (*DV-Konzept*) und
- auf Ebene einer IT-technischen *Implementierung*.

Sichten und Ebenen ergeben zusammen das so genannte *ARIS-Haus*, das in Abbildung 4.3 dargestellt ist.

ARIS unterstützt nur die Modellierung auf Ebene des *Fachkonzepts* und definiert dazu ein Metamodell (vgl. [Sch02, S.37]). Grafisch dargestellt werden die Sichten durch verschiedene Modelltypen, so zum Beispiel die *Steuerungssicht* durch (*erweiterte*) *ereignisgesteuerte Prozessketten (EPK)*. Die EPK ist der wohl bekannteste Modelltyp und wird gelegentlich synonym zu „ARIS“ verwendet (vgl. [Leh08, s.1]).

Mit dem Akronym „ARIS“ wird neben dem eigentlichen Modellierungsansatz auch eine ganze Produktgruppe von Modellierungswerkzeugen der Firma *Software*

AG [Sof] bezeichnet, die zum Teil mehr als die oben genannten Sichten durch Modelltypen unterstützen. Feiri et al. zählen in [FKKL09] 117 verschiedene Modelltypen im Werkzeug ARIS 7.02. Andere ARIS-Produkte wie *ARIS Express* und *arisccloud* verfügen über deutlich weniger Modelltypen. Selbst der in der Praxis verbreitete Modelltyp EPK ist in den Werkzeugen unterschiedlich realisiert. Auf Grund dieser Unterschiede wird die Modellierungssprache anhand der ARIS-Sichten und ihrer Metamodelle bewertet und nicht anhand der konkreten, aber werkzeughängigen Modelltypen. Gegenstand der Bewertung ist somit der „klassische“ ARIS Modellierungsansatz ohne Werkzeug-spezifische Erweiterungen. Die wesentlichen Sichten und ihre Elementtypen sind:

Datensicht: Umfelddaten, Nachricht

Leistungssicht: Leistung

Funktionssicht: Funktion, Anwendungssoftware

Organisationssicht: menschliche Arbeitsleistung

Steuerungssicht: Kombination aller oben genannten Elementtypen sowie zusätzlich Ereignis, Organisationseinheit, Kontrollfluss, Informationsfluss

Die ARIS Steuerungssicht rückt den logischen Ablauf von Geschäftsprozessen mit ihren Varianten in den Fokus. Abbildung 4.4 zeigt einen Ausschnitt eines Geschäftsprozesses, dargestellt als ereignisgesteuerte Prozesskette. Darin ist eine Sequenz von Tätigkeiten zu sehen, die zwei Varianten des Prozesses beschreibt: Eine, in der ein manueller Eingriff eines „Leistungsspezialisten“ nötig ist und eine, die ohne manuelle Tätigkeit durchgeführt wird. Akteure, Arbeitsgegenstände und dergleichen nehmen in einer solchen Algorithmus-artigen Sichtweise eine untergeordnete Rolle als „Ressourcen“ ein. Daher ermöglicht ARIS keine anwendungsorientierten Sichten auf die von einer Anwendungslandschaft unterstützten Geschäftsprozesse, denn in solchen Sichten stehen die handelnden Akteure mit ihren Aufgaben und Tätigkeiten im Vordergrund. Auch, dass Modelle vage sind, bleibt in einer von Ablauflogik geprägten Sicht unberücksichtigt.

Das in ARIS definierte Metamodell deckt nur die Ebene des Fachkonzepts ab und ermöglicht damit zwar einige fachliche, aber keine technologischen Sichten und keine fachlichen Architekturen. Die ARIS Modellierungswerkzeuge bieten jedoch

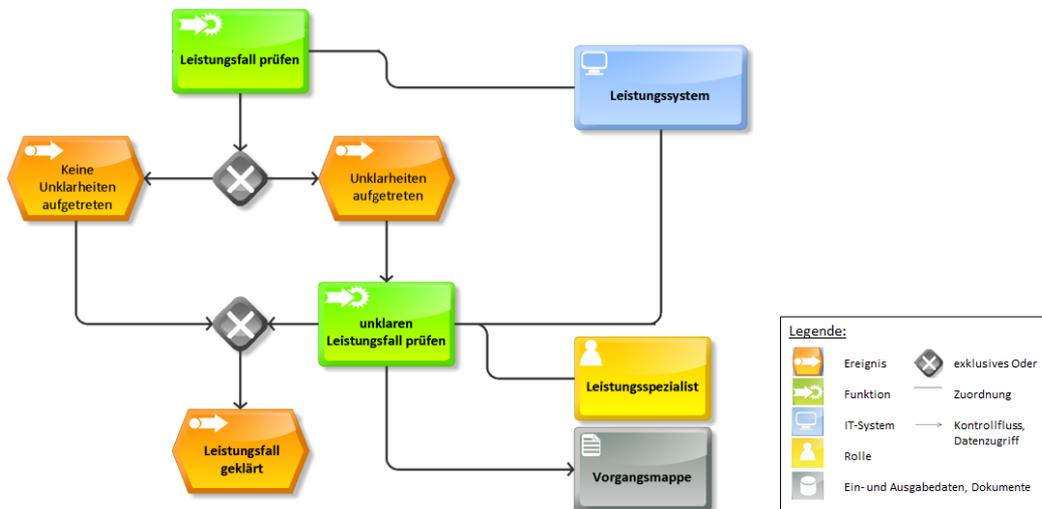


Abbildung 4.4: Ausschnitt eines Geschäftsprozesses, dargestellt als EPK (vollständige Grafik siehe Abbildung B.6)

zusätzliche Darstellungsmittel, die im Rahmen dieser Arbeit nicht auf ihre Eignung für Umgestaltungsprojekte analysiert wurden.

Das in [KNS92] beschriebene Vorgehen für die Gestaltung von Geschäftsprozessen und ihrer IT-Unterstützung sagt nichts über das Modellierungsvorgehen aus. Zwar wird gefordert, dass die Modelle validiert und auf Konsistenz geprüft werden sollen. Wie aber die Modelle zu einem Gestaltungsgegenstand werden, bleibt offen.

4.2.3 BEN

Der *Business Engineering Navigator (BEN)* der Universität St. Gallen wurde zur ingenieurmäßigen Veränderung von Unternehmen (*business engineering*) entwickelt. Er geht explizit auf die „Umgestaltung von Geschäftslösungen einschließlich deren IT-Unterstützung“ [Win10, S.6] ein.

Mit *ADOben* steht ein Werkzeug zur Verfügung, das auf dem kommerziellen Modellierungswerkzeug *ADOit 3.0* der *BOC Group* [BOCa] aufbaut. Die hier bewertete *ADOben* Version 1.0 und wurde 2009 veröffentlicht und ist mittlerweile kostenlos über die *Open Models Initiative* der Universität Wien [Ope] erhältlich.

BEN definiert ein Metamodell und teilt es in die Ebenen *Strategie*, *Organisation*, *IT/Business Alignment* und *IT-Ebene* ein. Statt eine Modellierungssprache für das gesamte Metamodell zu beschreiben, schlägt BEN verschiedene Modelltypen vor, die das Metamodell umsetzen sollen. Für einige diese Modelltypen werden in

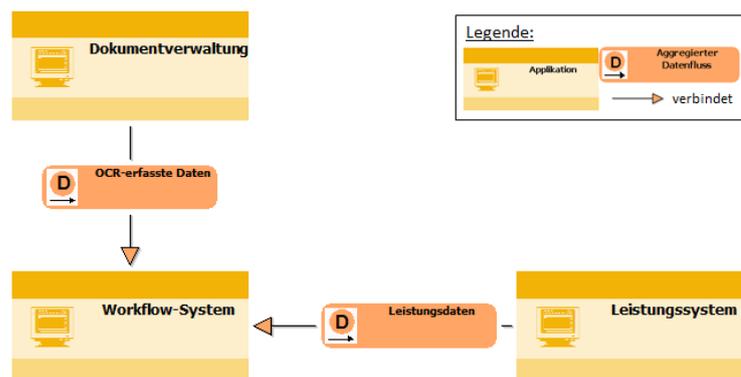


Abbildung 4.5: Ausschnitt einer fachlichen Architektur einer Anwendungslandschaft, dargestellt mit *ADOben* (vollständige Grafik siehe Abbildung B.13)

ADOben konkrete Notationen festgelegt. Bei anderen Modelltypen wird auf existierende Notationen wie z.B. EPK (siehe Abschnitt 4.2.2) zur Modellierung von Geschäftsprozessen verwiesen, statt einen eigene Notation vorzuschlagen. *ADOben* unterstützt daher nur einen Teil des BEN Ansatzes und weicht zudem in der Benennung der Modelltypen und Modellelementtypen von den in der Literatur zu BEN verwendeten Begriffen ab. Die Beispielmuster in Abschnitt B.3 zeigen nur die mit *ADOben* erstellbaren Modelle. Die Bewertung wird anhand des BEN Metamodells und der vorgeschlagenen Modelltypen vorgenommen. Die wesentlichen Modelltypen und ihre Modellelementtypen sind:

Applikationslandschaft oder -landkarte: Applikation, fachlicher Service, logisches Datenobjekt

Ablaufmodell: Aufgabe, Rolle

Informationsmodell oder -landkarte: Aufgabe, Geschäftsobjekt

Durch das Metamodell gibt BEN vor, welche Modellierungssprachen zur Ausgestaltung geeignet sind. Anwendungsorientierung und Ausdrucksmittel, um Zweifel an der Brauchbarkeit von Modellen auszudrücken, gehören nicht zu den Vorgaben. Daher überrascht es nicht, dass die in BEN beschriebenen und in *ADOben* implementierten Modellierungssprachen nicht anwendungsorientiert sind und vages Modellieren nicht ermöglichen. Trotzdem können mit dem in BEN definierten Metamodell fachliche Sichten und fachliche Architekturen (siehe Abbildung 4.5) abgedeckt werden. Technologische Sichten werden hingegen nicht berücksichtigt.

BEN geht auf den Modellierungsprozess ein, weil Vorgehensmodelle für vier verschiedene Projekttypen vorgestellt werden. Diese Projekttypen können als Umgestaltungsprojekte interpretiert werden. Für jeden Projekttyp wird zwar festgelegt, in welcher Reihenfolge die verschiedenen Modelltypen eingesetzt werden sollen, aber nicht, wie die konkreten Modelle erstellt werden. Die Vorgehensmodelle gehen auch nicht darauf ein, ob und wie Modelle rückgekoppelt werden sollen.

4.2.4 EAM Patterns

Die *EAM Patterns* aus dem *Enterprise Architecture Management Pattern Catalog* [BELM08] der TU München sind eine Zusammenstellung von Problemstellungen (sogenannten *Concerns*) und passenden Lösungsmustern für das Management von IT-Unternehmensarchitekturen (siehe Abschnitt 1.2). Die Muster beschreiben „best practices“ für Analyseverfahren, Darstellung und Informationsmodelle und sind als Ergänzung anderer Ansätze gedacht. Beispielsweise bestehen die Darstellungsmuster aus Viewpoints, in denen UML-Modelle (vgl. [RJB05]) und Software-Karten (vgl. [MW04]) eingesetzt werden. Die Darstellungsmuster definieren also keine eigene Modellierungssprache, sondern setzen etablierte Sprachen und „freie“, durch Legenden erläuterte Darstellungen ein. Einige der Darstellungsmuster verwenden eine Modellierungssprache, die große Ähnlichkeit zu ArchiMate hat (vgl. Abschnitt 4.2.1). Die für die Bewertung wesentlichen Muster sind:

Einsatzszenarien (*Concerns*): siehe [BELM08, S.32–36]

C–9: „Possibilities to reorganize the application landscape [...] should be outlined.“

C–34: „How does the long-term vision, the target of the application landscape, look like?“

C–35: „How does the application landscape look like at a specific date?“

C–36: „Which dependencies exist between business applications and are affected by current or planned projects?“

C–78: „To which extent are the business processes supported by business applications?“

Vorgehensmuster (*M-Patterns*): siehe [BELM08, S.64–74]

M-14: „Development of planned and target landscapes“

M-15: „Management of the application lifecycle“

M-29: „High level process support“

M-30: „Business Process Data Flow Analysis“

Darstellungsmuster (V-Patterns): siehe [BELM08, S.107–168]

V-18: „Service-based Business Process Support Map“

V-47: „Business Object Class Diagram“

V-48: „Cluster Map visualizing Business Object Flows between Business Applications“

V-51: „Process Overview“

V-52: „Business-level Communication Overview“

V-66: „Architectural Solution in Detail“

Die Darstellungsmuster finden sich in verschiedenen Modellierungswerkzeugen wieder. Es gibt jedoch kein Werkzeug, das speziell auf den EAM Pattern Catalog zugeschnitten ist. Die Beispiele in Anhang B.4 wurden nicht mit einem Modellierungswerkzeug, sondern mit Microsoft PowerPoint erstellt.

Der EAM Musterkatalog adressiert mit seinen „Concerns“ viele Fragestellungen, die für Umgestaltungsprojekte relevant sind. Die Darstellungsmuster beschreiben technologischen Sichten, fachliche Architekturen und zum Teil auch fachliche Sichten. Anwendungsorientierte Darstellungsmuster finden sich nicht im Musterkatalog.

Der Modellierungsprozess wird in den EAM Patterns nicht berücksichtigt, da die Modelle als Visualisierung zuvor gesammelter Informationen aufgefasst werden. Ebenso wenig wird darauf eingegangen, dass Modelle vage sind. In der Sichtweise der Autoren der EAM-Patterns werden Modelle zwar zur Kommunikation eingesetzt, Modellierung selbst ist aber kein Lern- und Kommunikationsprozess. Die Unschärfe von Informationen ist in dieser Sichtweise kein Problem der Modellierung, sondern der Informationsbeschaffung, die von den EAM-Patterns nicht thematisiert wird.

4.2.5 eGPM

Die *exemplarische Geschäftsprozessmodellierung* (eGPM, vgl. [BH12]) basiert auf sogenannten *Kooperationsbildern* (vgl. [KWR96]), die an der Universität Hamburg

zur Darstellung von kooperativen Arbeitsabläufen entwickelt wurden. Ergänzt um ein Szenario-basiertes Konzept (vgl. [Car00]) und weitere Modelltypen wurde die eGPM zur Anforderungsanalyse in der objektorientierten Anwendungsentwicklung eingesetzt.

Im Fokus der eGPM liegen menschliche Akteure und wie ihre Aufgaben durch Softwaresysteme unterstützt werden. In der Regel werden mit eGPM nur einige typische, exemplarische Verläufe eines Prozesses modelliert und nicht alle möglichen Varianten. Dieses Vorgehen war namensgebend für den Modellierungsansatz - *exemplarische* Geschäftsprozessmodellierung. Unter Mitarbeit des Autors wurde der Ansatz weiterentwickelt und um ein Darstellungsmittel für Anwendungslandschaften – den Modelltyp *IT-Landschaft* – ergänzt.

Das Modellierungswerkzeug für die eGPM basiert auf der kommerziellen Metamodellierungsplattform *ADONIS* der *BOC Group* ([BOCa]). Eine kostenfreie Version des Modellierungswerkzeugs mit reduzierter Funktionalität wird über die *Open Models Initiative* der Universität Wien [Ope] bereitgestellt. Die hier bewertete Version des Ansatzes und des zur Modellierung der Beispiele verwendeten Werkzeugs tragen die Bezeichnung *2014.10.08*. Die wesentlichen Modelltypen und Modellelementtypen sind:

Kooperationsbild: Akteur, Gegenstand, Aggregation, informiert mit (Relation), gibt Gegenstand weiter (Relation), bearbeitet (Relation)

IT-Landschaft: System, Daten, holt Daten (Relation), steht in Verbindung mit (Relation)

Begriffsmodell: Dokument/Unterlage, gehört zu (Relation)

Die Stärke eGPM liegt in der Darstellung von anwendungsorientierten Sichten auf die Geschäftsprozesse einer Anwendungslandschaft. Abbildung 4.6 zeigt, wie die Akteure „Posteingang“ und „Sachbearbeiter“ über ein Softwaresystem „Dokumentverwaltung“ zusammenarbeiten. Auch weitere fachliche Sichten und fachliche Architekturen können mit der eGPM modelliert werden. Für technologische Sichten bietet sie hingegen keine geeigneten Darstellungsmöglichkeiten.

Durch das Szenario-basierte Modellierungsvorgehen erzwingt die eGPM, bewusst Annahmen zu treffen und unvollständig zu modellieren. Damit einher geht ein Modellierungsvorgehen in Form von Modellierungsworkshops, die Modellie-

und der von ihnen unterstützten Geschäftsprozesse, aber trotzdem erfüllen sie nicht alle, im Kontext von Umgestaltungsprojekten auftretenden Anforderungen. Allerdings zeigt die Bewertung, dass einige Ansätze über Mittel verfügen, die sich für Umgestaltungsprojekte eignen. Im nächsten Kapitel werden solche Mittel identifiziert, und daraus Bausteine entwickelt, mit denen Modellierungsansätze besser an Umgestaltungsprojekte angepasst werden können.

Kapitel 5

Ein Modellierungsansatz für Umgestaltungsprojekte

Ziel dieses Kapitels ist, die Leser in die Lage zu versetzen, Modellierungsansätze an Umgestaltungsprojekte anzupassen. Dazu wird zunächst beschrieben, wie Umgestaltungsprojekte idealerweise unterstützt werden. Anschließend werden sogenannte „Bausteine“ vorgestellt, die dabei helfen, dem Ideal näher zu kommen. Es ist naheliegend, die Bausteine aus den Erkenntnissen aus Kapitel 4 abzuleiten und bewährte Sprachmittel und Vorgehensweisen der bewerteten Ansätze wiederzuverwenden.

Im dritten Teil dieses Kapitels wird die Anwendung der Bausteine gezeigt. Ein kommerzieller Modellierungsansatz wird für Umgestaltungsprojekte angepasst.

5.1 Zielvorstellung: Ein idealer Modellierungsansatz

Um besser an Umgestaltungsprojekte angepasste Modellierungsansätze zu entwickeln, braucht man eine Zielvorstellung. Die bewerteten Modellierungsansätze eignen sich dazu nicht, da keiner von ihnen alle Anforderungen erfüllt. Daher wird in diesem Abschnitt skizziert, wie ein Umgestaltungsprojekt ideal durch einen Modellierungsansatz unterstützt wird:

Ein idealer Modellierungsansatz für Umgestaltungsprojekt macht sichtbar, wie die Anwendungslandschaft die Geschäftsprozesse unterstützt. Dazu müssen die Geschäftsprozessmodelle zeigen, welche Tätigkeiten, Arbeitsgegenstände, Arbeitsergebnisse und Informationen durch welche Softwaresysteme realisiert werden. Das Modellierungswerkzeug des Ansatzes ermöglicht, aus einem Geschäftsprozessmo-

dell in die Modelle der fachlichen Architektur und der Bausteinsicht zu wechseln, in denen die IT-Unterstützung der Geschäftsprozesse dargestellt wird. Dadurch kann zum Beispiel die Frage beantwortet werden, ob eine Tätigkeit von der existierenden Anwendungslandschaft unterstützt werden kann oder ob dazu erst neue Schnittstellen zwischen Softwaresystemen geschaffen werden müssen.

Die mit einem idealen Ansatz erstellten Modelle erleichtern den Akteuren, ihre Aufgaben und Tätigkeiten in den Modellen wiederzuerkennen. Ein solcher Ansatz versetzt die Akteure in die Lage, sich direkt an der Modellierung zu beteiligen und die Modelle zu validieren, während sie entstehen. Die Person, die den Modellierungsprozess moderiert und in Modellen festhält, kann die Rückmeldung der Akteure mit Hilfe des Modellierungswerkzeugs unmittelbar in die Modelle einfließen zu lassen.

Der Ansatz ermöglicht, pragmatisch zu modellieren: Für das Umgestaltungsziel wichtige Zusammenhänge können erkannt und durch mehrere Modelle aus verschiedenen Blickwinkeln dargestellt werden. Unvollständige, widersprüchliche und nur grob modellierte Informationen können als solche gekennzeichnet werden, sodass die Modellnutzer entscheiden können, wie brauchbar ein Modell für einen bestimmten Zweck ist.

Die Modellierer werden bei ihren anspruchsvollen Tätigkeiten durch den Ansatz unterstützt: Der Ansatz hilft dabei, für den Modellierungszweck geeignete Modelltypen zu wählen. Außerdem beschreibt der Ansatz, welche Fragestellungen durch welche Modellanalysen und Modellmanipulationen beantwortet werden können.

Neben den speziellen Anforderungen, die sich in Umgestaltungsprojekten an einen Modellierungsansatz ergeben, soll dieser auch dabei unterstützen, qualitativ hochwertige Modelle zu erstellen. Allgemeine Qualitätsmerkmale für Modelle werden beispielsweise von Mohagheghi et al. in [MDN08] beschrieben und umfassen unter anderem die Verständlichkeit von Modellen. Solche Qualitätsmerkmale werden im Gebiet der konzeptuellen Modellierung schon lange diskutiert und liegen nicht im Fokus dieser Arbeit. Deswegen wird nicht weiter auf allgemeine Qualitätsmerkmale von Modellen eingegangen.

5.2 Bausteine eines Modellierungsansatzes

Will ein Unternehmen ein Umgestaltungsprojekt durch Modellierung unterstützen, wird es einen Ansatz wählen, der die Spezifika der Branche und des Unternehmens

abbilden kann. Auch bereits getätigte Investitionen in Modellierungswerkzeuge und die Ausbildung von Modellierern spielen eine Rolle bei der Wahl eines geeigneten Modellierungsansatzes. Daher hat diese Arbeit nicht zum Ziel, *den einen, idealen* Modellierungsansatz für Umgestaltungsprojekte zu entwickeln. Vielmehr will sie dabei unterstützen, bestehende Modellierungsansätze für solche Projekte anzupassen. Dieser Abschnitt stellt zehn „Bausteine“ vor. Jeder Baustein hilft dabei, eine oder mehrere Anforderungen aus dem Anforderungskatalog (siehe Kapitel 3) zu erfüllen. Als Vorlage für die Bausteine dienen geeignete Sprachmittel und Vorgehensweisen der zuvor bewerteten Modellierungsansätze. Um die Bausteine außerhalb ihres ursprünglichen Ansatzes verwenden zu können, müssen die Sprachmittel und Vorgehensweisen generalisiert werden, wie folgendes Beispiel zeigt:

- Vorlage für Baustein: Der Modelltyp „Kooperationsbild“ der eGPM eignet sich gut für Umgestaltungsprojekte, unter anderem weil er anwendungsorientierte Aspekte von Geschäftsprozessen darstellen kann.
- Baustein: Einer der Bausteine beschreibt die für Umgestaltungsprojekte relevanten Merkmale des Kooperationsbildes, ohne dabei auf die konkrete Modellierungssprache einzugehen.
- Anwendung des Bausteins: Jeder Modelltyp zur Geschäftsprozessmodellierung kann durch den Baustein so angepasst werden, dass anwendungsorientierte Aspekte dargestellt werden kann.

Die Bausteine machen also Vorgaben an Modellierungssprachen und Vorgehensweisen, statt konkrete Sprachen und Vorgehensweisen zu definieren. Die Bausteine sollen dadurch als Blaupause dienen, um existierende Ansätze besser an Umgestaltungsprojekte anzupassen. Angesichts der Zahl bereits bestehender Ansätze ist dieses Ziel praxisrelevanter als einen ganz neuen Modellierungsansatz zu entwickeln. Für letzteres reichen die Bausteine nicht aus, weil dazu weitere Voraussetzungen nötig sind, die nicht im Rahmen dieser Arbeit entwickelt werden:

- Eine Modellierungssprache mit definierter Notation, Syntax und Semantik.
- Attribute zur Beschreibung der Modellelemente.
- Eine Modellierungswerkzeug.

Baustein	für Bestandteil
1 Modelltypen zur Darstellung von Geschäftsprozessen	Modellierungssprache
2 Modelltyp zur Darstellung der fachlichen Architektur einer Anwendungslandschaft	Modellierungssprache
3 Modelltyp zur Darstellung von fachlichen Konzepten und ihrer Zusammenhänge	Modellierungssprache
4 Modelltyp zur Darstellung einer Bausteinsicht auf Anwendungslandschaften	Modellierungssprache
5 Vage Informationen grafisch im Modell kennzeichnen	Modellierungssprache
6 Modelle im Werkzeug miteinander verknüpfen, wenn sie dasselbe Original in verschiedenen Sichten darstellen	Modellierungssprache
7 Der modellierte Gegenstandsbereich wird als Abbildung einer existierenden Anwendungslandschaft (Ist-Modell) oder als Modell eines zukünftigen Soll-Zustands gekennzeichnet	Modellierungssprache
8 Szenario-basiertes Modellieren von Geschäftsprozessen	Modellierungssprache Modellierungsvorgehen
9 Beteiligte aus unterschiedlichen Aufgabengebieten modellieren gemeinsam unter Anleitung eines Moderators	Modellierungsvorgehen
10 Beschreibung, welcher Zweck mit welchen Modeltypen erreicht werden kann	Umgang mit Modellen

Tabelle 5.1: Bausteine eines Modellierungsansatzes für Umgestaltungsprojekte

Das Vorgehen bei der Entwicklung der Bausteine wurde in Abschnitt 1.3 beschrieben. Tabelle 5.1 gibt einen Überblick über die Bausteine, die im Folgenden hergeleitet werden:

Die bewerteten Ansätze ermöglichen, mit ihren Modelltypen Sichten auf Anwendungslandschaften darzustellen. Die Bausteine 1–4 beschreiben, mit welchen Modelltypen vier, für Umgestaltungsprojekte relevante Sichten realisiert werden können:

Baustein 1. Modelltyp zur Darstellung von Geschäftsprozessen.

Der Modelltyp muss in der Lage sein, anwendungsorientierte Zusammenhänge auszudrücken, um die Anforderungen 1–3 erfüllen zu können. Es muss dargestellt werden können, welche Werkzeuge, Arbeitsgegenstände, Arbeitsergebnisse und Informationen über Arbeitsabläufe durch Softwaresystemen realisiert werden. Das heißt:

- Arbeitsgegenstände und -ergebnisse müssen grafisch mit Softwaresystemen in Beziehung gesetzt werden können.
- Manuelle und IT-gestützte Tätigkeiten müssen unterscheidbar sein.
- Es muss darstellbar sein, wie Akteure miteinander kommunizieren und Arbeitsabläufe koordinieren.
- Aktive Elemente (Akteure, Softwaresysteme) und passive Elemente (z.B. Arbeitsgegenstände und -ergebnisse) eines Geschäftsprozesses müssen im Modell unterschieden werden können.

Die Modelltypen *Kooperationsbild* (eGPM) und – zum Teil – *Business Process Viewpoint* (ArchiMate) können als Vorlage für einen geeigneten Modelltyp zur Darstellung von Geschäftsprozessen dienen.

Baustein 2. Modelltyp für fachliche Architekturen von Anwendungslandschaften.

Der Modelltyp muss in der Lage sein, eine Brücke zwischen fachlichen und technologischen Sichten zu schlagen, um die Anforderung 5 erfüllen zu können. Dazu muss dargestellt werden können, welche fachlichen Zusammenhänge es zwischen den Softwaresystemen einer Anwendungslandschaft gibt.

Das EAM-Pattern *V-48* und der Modelltyp *IT-Landschaft* der eGPM können als Vorlage für einen geeigneten Modelltyp für fachliche Architekturen dienen.

Baustein 3. Modelltyp für fachliche Konzepte und ihrer Zusammenhänge.

Der Modelltyp muss die Beziehungen zwischen Arbeitsgegenständen und Arbeitsergebnissen ausdrücken können, um die Anforderung 4 zu erfüllen.

Der *Information Structure Viewpoint* aus ArchiMate und der Modelltyp *Begriffsmodell* der eGPM können als Vorlage für einen geeigneten Modelltyp für fachliche Konzepte dienen.

Baustein 4. Modelltyp für eine Bausteinsicht auf Anwendungslandschaften.

Der Modelltyp muss Folgendes ausdrücken können, um Anforderung 6 zu erfüllen:

- Die Softwaresysteme und ihre Kopplung mit anderen Softwaresystemen.
- Den inneren Aufbau der Softwaresysteme.

Die ArchiMate-Viewpoints *Application Cooperation Viewpoint* und *Application Structure Viewpoint* und das EAM-Pattern *V-66* können als Vorlage für einen geeigneten Modelltyp für Bausteinsichten dienen.

In Abschnitt 2.1 wurde beschrieben, was eine anwendungsorientierte Perspektive auf Anwendungslandschaften ausmacht. Veranschaulicht wurde dies durch Abbildung 2.5. Zeichnet man in diese die Bausteine 1–4 ein, erhält man Abbildung 5.1. Sie zeigt, welche anwendungsorientierten Aspekte die in den Bausteinen beschriebenen Modelltypen darstellen müssen. Die Abbildung hilft dabei, die Modelltypen zu einem Ansatz zu integrieren, der alle geforderten Sichten auf eine Anwendungslandschaft darstellen kann.

Baustein 5. Vage Informationen grafisch im Modell kennzeichnen.

Modellierer brauchen eine Möglichkeit, um „Zweifel über die Angemessenheit oder Vollständigkeit“ [HHL99, S.71] von Modellen auszudrücken. Die eGPM zeigt, wie durch ein Attribut „offene Fragen“ Modellelemente als vage gekennzeichnet werden können (siehe Bewertung von Anforderung 8).

Baustein 6. Modelle im Werkzeug miteinander verknüpfen, wenn sie dasselbe Original in verschiedenen Sichten darstellen.

Ein Modellierungsansatz soll den Modellbenutzern vergegenständlichen, welche Modelle miteinander in Beziehung stehen und gegebenenfalls gemeinsam betrachtet und analysiert werden müssen. Abbildung 5.1 zeigt, welche anwendungsorientierten

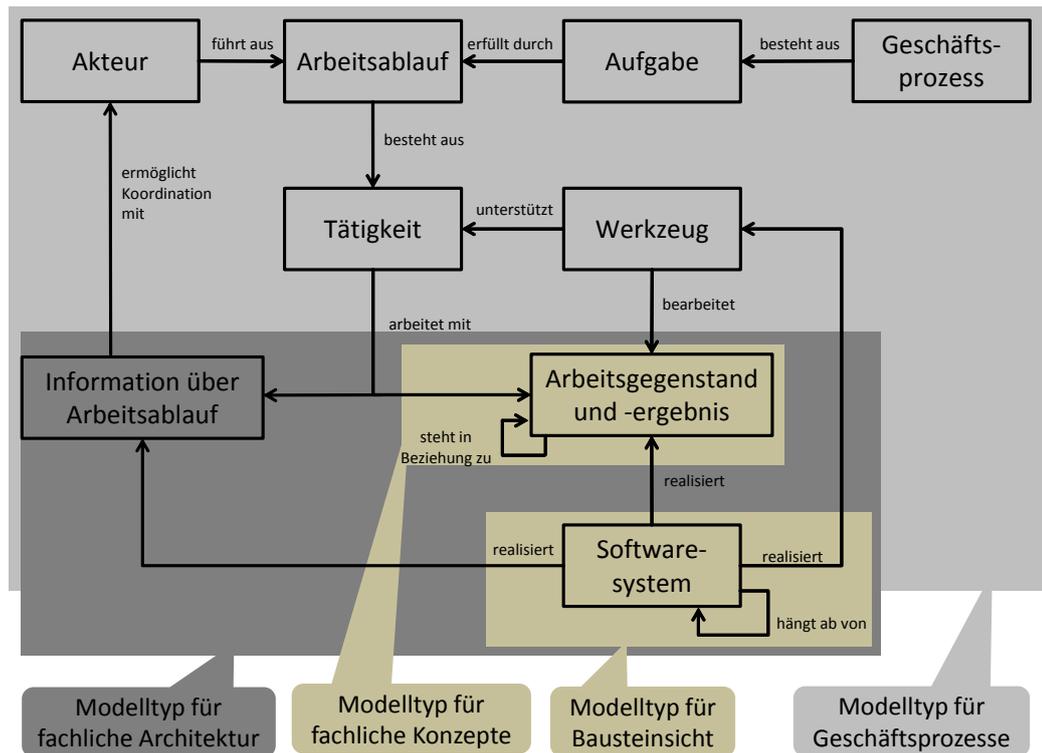


Abbildung 5.1: Modelltypen eines Modellierungsansatzes für Umgestaltungsprojekte

Aspekte in mehreren Modelltypen vorkommen. Sie können Modelle verschiedener Typen „verbinden“. Zwei Beispiele dafür sind:

- Arbeitsgegenstände und Arbeitsergebnisse verbinden Geschäftsprozessmodelle, fachliche Architekturen und Modelle von fachlichen Konzepten.
- Softwaresysteme verbinden fachliche Architekturen und Modelle der Bausteinsicht.

Die Metamodelle aller bewerteten Ansätze ermöglichen solche Verknüpfungen und viele Modellierungswerkzeuge implementieren sie als Referenzen, die einen Wechsel in ein verknüpftes Modell ermöglichen (siehe Bewertung Anforderung 7).

Baustein 7. Kennzeichnung als Ist- oder Soll-Modell.

Mit Anforderung 13 wurde gefordert, dass Modelle als Ist- oder Soll-Modelle kennzeichnenbar sein müssen. Als Beispiel, wie diese Anforderung erfüllt werden kann, dient die eGPM. Sie bietet auf Modellebene ein Attribut, dass Modelle als Ist- oder Soll-Modell kennzeichnet.

Baustein 8. Szenario-basiertes Modellieren von Geschäftsprozessen.

Diese Modellierungstechnik trägt dazu bei, Anforderung 10 (vages Modellieren) zu erfüllen. Szenario-basiertes Modellieren von Geschäftsprozessen ist eine im Requirements Engineering etablierte Technik, die zum Beispiel in *Use Cases* (siehe u.a. [JEJ95] und [Coc00]) zum Einsatz kommt. Szenarien werden in Use Cases eingesetzt, um zu beschreiben, wie sich ein Softwaresystem im Regelfall, in wichtigen Sonderfällen und in Fehlerfällen verhalten soll. In [Car00, S.46f] beschreibt Carroll Szenarien folgendermaßen:

- Szenarien sind Geschichten über Menschen und ihre Tätigkeiten.
- Szenarien handeln in einem festgelegten Kontext.
- Szenarien enthalten Akteure, die typischerweise Ziele und Vorgaben haben.
- Szenarien haben eine Handlung. Sie bestehen aus einem Ablauf von Tätigkeiten und Ereignissen.

Weil jedes Szenario-basierte Modell *einen* konkreten Verlauf eines Geschäftsprozesses darstellt, können Szenarien von den Akteuren gegen ihre tatsächlichen Arbeitsabläufe abgeglichen und die Modelle auf diesem Weg validiert werden. Szenarien unterstützen damit die Rückkopplung im Modellierungsprozess, welche für Eigenschaft 1 (Subjektive Brauchbarkeit), Eigenschaft 2 (Gestaltungsgegenstand) und Eigenschaft 3 (Lern- und Kommunikationsprozess) erforderlich ist.

Weil beim Szenario-basierten Modellieren keine Sprachmittel für Fallunterscheidungen und Schleifen benötigt werden, sind „klassische“ Modellierungssprachen für Geschäftsprozesse kein optimales Mittel, um Szenarien darzustellen. Daher schreibt dieser Baustein nicht nur ein Modellierungsvorgehen vor, sondern verlangt, dass der im Baustein 1 geforderte Modelltyp für Geschäftsprozesse das Szenario-basierte Modellieren unterstützt. Als Beispiel dafür dient die eGPM mit dem Modelltyp „Kooperationsbild“ und einem darauf abgestimmten, zyklischen Modellierungsvorgehen. In den Zyklen werden gemeinschaftlich Kriterien festgelegt, nach denen entschieden wird, welche Szenarien relevant sind und welche nicht. Dadurch wird auf Modellebene sichtbar, dass Geschäftsprozesse nicht vollständig erfasst sind.

Baustein 9. Beteiligte aus unterschiedlichen Aufgabengebieten modellieren gemeinsam unter Anleitung eines Moderators.

Aufgabe	betroffene Rollen
Informationen erfassen	Modellierer
Bewerten und Entscheiden	Projektleitung, Architekt
Planen	Projektleitung, Architekt
Einbinden der Organisation	alle Projektmitarbeiter, Linienorganisation
Technische Umsetzung	Tester, Software-Entwickler

Tabelle 5.2: Aufgaben und Rollen, die durch den Modellierungsansatz unterstützt werden sollen

Dieser Baustein ermöglicht, verschiedene Einschätzungen des zu modellierenden Sachverhalts zu diskutieren. Dabei können Widersprüche erkannt und beseitigt werden. Somit werden Modelle während der Modellierung validiert (Anforderung 9). Die subjektive Brauchbarkeit von Modellen wird für die Beteiligten erfahrbar (Eigenschaft 1). Außerdem kann gemeinschaftliches Modellieren einen Lern- und Kommunikationsprozess etablieren (Eigenschaft 3). Gemeinschaftliches Modellieren ist ein wesentlicher Bestandteil des Modellierungsvorgehens der eGPM, aus der dieser Baustein abgeleitet wurde.

Baustein 10. Beschreibung, welcher Zweck mit welchen Modeltypen erreicht werden kann.

Um konkrete Einsatzszenarien für einen Ansatz zu entwickeln (siehe Anforderung 11), muss festgelegt werden, wer welche Modeltypen wozu einsetzen soll. Ein Beispiel dafür liefert ArchiMate. Um aus diesem Beispiel einen Baustein abzuleiten, können die Rollen und Aufgaben in Umgestaltungsprojekten herangezogen werden (siehe Abschnitt 2.3). Modelle haben den Zweck, diese Aufgaben zu unterstützen. Tabelle 5.2 zeigt, welche Aufgaben und Rollen unterstützt werden sollen. Dieser Baustein wird umgesetzt, indem man die Modeltypen eines Modellierungsansatzes den Aufgaben und Rollen aus Tabelle 5.2 zuordnet.

Jeder der zehn Bausteine hilft, eine oder mehrere Anforderungen an einen Modellierungsansatz für Umgestaltungsprojekte zu erfüllen. Bis auf Anforderung 12 (Modellmanipulation und -analyse) werden alle Anforderungen abgedeckt. Wie in Abschnitt 3.2 beschrieben, machen Modellierungswerkzeuge den Umgang mit Modellen oft erst lohnenswert. Abstrahiert von konkreten Modellierungssprachen und -werkzeugen ist es daher wenig hilfreich, zu beschreiben, wie man Modelle für bestimmte Aufgaben manipulieren und analysieren soll.

Die Bausteine stammen aus praxiserprobten Ansätzen und haben sich in ihrem ursprünglichen Kontext bewährt. Jedoch sind noch weitere Möglichkeiten denkbar, mit denen die Anforderungen erfüllt werden können. Im Ausblick dieser Arbeit (siehe Kapitel 7) wird dies thematisiert.

Ob ein Ansatz, der mit den Bausteinen angepasst wurde, die Anforderung erfüllt, hängt nicht nur von den Bausteinen selbst ab. Auch die Qualität der Anpassung ist relevant. Der nächste Abschnitt beschreibt beispielhaft, wie ein Modellierungsansatz mit den Bausteinen angepasst werden kann.

5.3 Anwendungsbeispiel

In diesem Abschnitt wird anhand eines Beispiels gezeigt, wie die Bausteine eingesetzt werden, um einen Modellierungsansatz anzupassen. Dazu muss zuerst ein zu adaptierender Modellierungsansatz ausgewählt werden.

5.3.1 Auswahl des zu adaptierenden Modellierungsansatzes

Der Modellierungsansatz wurde für diese Arbeit nach folgenden Gesichtspunkten ausgewählt:

- Da die Bausteine keine konkrete Modellierungssprache definieren, muss der Ansatz bereits grundsätzlich für die Darstellung von Anwendungslandschaften und ihres Einsatzes in Geschäftsprozessen geeignet sein.
- Das Anwendungsbeispiel soll zeigen, dass alle Bausteine außerhalb ihres ursprünglichen Kontextes anwendbar sind. Dazu muss ein Modellierungsansatz ausgewählt werden, der nicht für die Herleitung der Bausteine analysierte wurde.
- Das Anwendungsbeispiel muss nach denselben Kriterien bewertet werden können wie die Ansätze in Kapitel 4. In diese Bewertung wurden nur Ansätze mit kommerziellen Modellierungswerkzeugen einbezogen. Um die Bewertungsergebnisse aus Kapitel 4 mit denen des Anwendungsbeispiels vergleichbar zu machen, muss der auszuwählende Ansatz ebenfalls über ein kommerziell vertriebenes Modellierungswerkzeug verfügen.

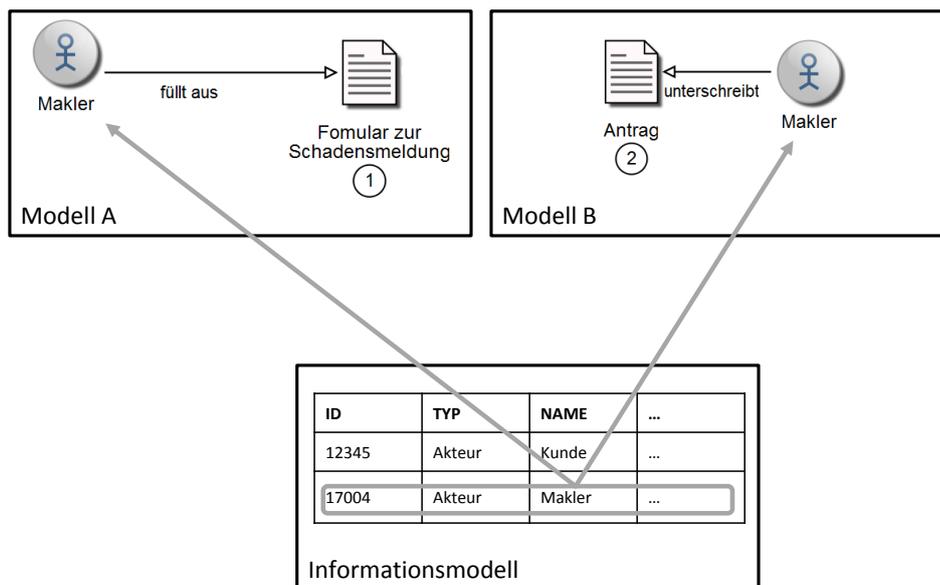


Abbildung 5.2: Das Modellelement „Makler“ in einem Informationsmodell und zwei grafischen Modellen

- Damit die vom Werkzeug unterstützte Modellierungssprache durch die Bausteine ergänzt werden kann, muss das zur Sprachdefinition verwendete Metamodel des Werkzeugs anpassbar sein.

Im Rahmen der beruflichen Tätigkeit des Autors stand mit *ADOit 6.0* der *BOC Group* [BOCa] ein werkzeuggestützter Modellierungsansatz zur Verfügung, der den oben genannten Gesichtspunkten entspricht. *ADOit* ist am Markt etabliert (vgl. [For13]) und auch als kostenfreie Version („community edition“, siehe [BOCb]) erhältlich.

Im Auslieferungsumfang von *ADOit* ist ein vordefiniertes Metamodel enthalten. Dieses wird üblicherweise im Rahmen der Werkzeugeinführung von der *BOC Group* gemeinsam mit der einsetzenden Organisation angepasst. Die Modellelemente der Modelle können in einem Datenbank-basierten Informationsmodell gespeichert werden (siehe Abbildung 5.2). Auf das Informationsmodell können Abfragen ausgeführt werden – beispielsweise, in welchen Modellen ein Modellelement visualisiert wird.

Modelle können in *ADOit* mit einem grafischen Editor erstellt werden. Zusätzlich ist es möglich, Modelle zu generieren, welche die Daten des Informationsmodells anhand vordefinierter und manuell erstellbarer Viewpoints (vgl. Definition 3.1.4) visualisieren.

5.3.2 Überblick über den Modellierungsansatz

Der beispielhafte Ansatz umfasst vier Modelltypen: Das *Kooperationsszenario*, die *Datenarchitektur*, die *Anwendungsarchitektur* und die *Technologiearchitektur*. Diese Modelltypen werden im Folgenden anhand eines Beispiels aus der Versicherungsbranche vorgestellt.

Wie die Bausteine umgesetzt wurden und welchen Beitrag der Autor dieser Arbeit daran hatte, wird im Abschnitt 5.3.3 im Detail beschrieben.

Kooperationsszenario

Modelle des Typs *Kooperationsszenario* stellen exemplarische Arbeitsabläufe dar und zeigen, wie menschliche *Akteure* miteinander und mit *Anwendungen* kooperieren und *Arbeitsgegenstände* bearbeiten. Kooperationsszenarien unterstützen unter anderem dabei, eine Anwendungslandschaft zu bewerten und geplante Veränderungen an die betroffenen Akteure zu kommunizieren.

Abbildung 5.3 zeigt, wie ein Versicherungskunde mit Hilfe eines Maklers einen Schadensfall an seine Versicherung meldet. Nachdem das entsprechende Formular ausgefüllt wurde, trägt der Makler im zweiten Schritt die Schadensmeldung zusätzlich in ein Softwaresystem *CRM*¹ ein. Dass diese Tätigkeit IT-gestützt ist, wird durch ein Symbol in der linken unteren Ecke visualisiert. Das Prozessmodell macht deutlich, dass die Daten zum Schadensfall doppelt (auf Papier und in einem Softwaresystem) erfasst werden müssen.

Nachdem die Schadensmeldung von einem Sachbearbeiter bearbeitet wurde, geht im letzten Schritt eine Regulierungsmeldung per Post an den Kunden. Ein Fragezeichen-Symbol in der rechten unteren Ecke visualisiert, dass es zu diesem Schritt noch offene Fragen gibt. Im Modellierungswerkzeug ist dazu ein Kommentar hinterlegt: „Wird der Makler von der Versicherung informiert?“.

Das Kooperationsszenario in Abbildung 5.3 zeigt den Prozessverlauf des positiven Falls – d.h., die Versicherung kommt für den Schaden des Kunden auf. Dieser Prozessverlauf wird offensichtlich schlecht von der Anwendungslandschaft des Versicherers unterstützt. Ob das einzige dabei eingesetzte Softwaresystem „CRM“ zum Beispiel mit Daten über die Schadensregulierung versorgt werden kann, lässt sich mit dem Modelltyp „Anwendungsarchitektur“ analysieren. Das Modellierungs-

¹Customer Relationship Management

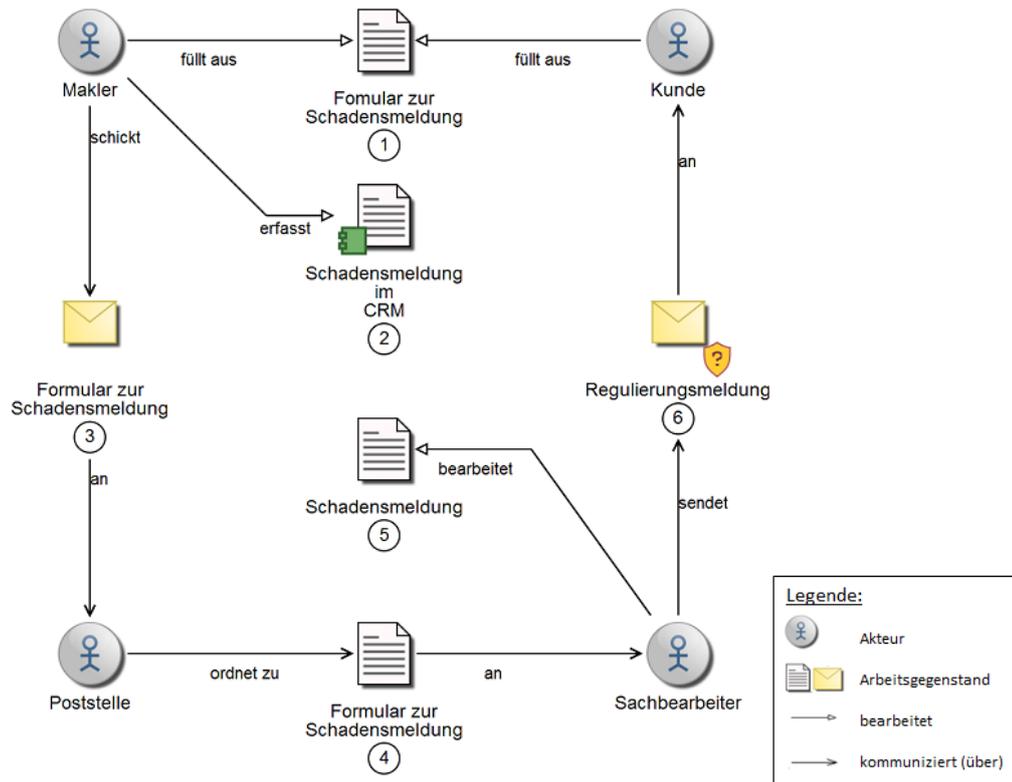


Abbildung 5.3: Beispiel für den Modelltyp Kooperations Szenario

werkzeug unterstützt dabei, aus dem Kooperations Szenario in Modelle des Typs „Anwendungsarchitektur“, in denen das CRM vorkommt, zu wechseln. Wie die im Prozess eingesetzten Arbeitsgegenstände und die erarbeiteten Arbeitsergebnisse zusammenhängen, kann durch Modelle des Typs „Datenarchitektur“ veranschaulicht werden. Das Werkzeug unterstützt, aus Kooperations Szenarien in relevante Datenarchitektur-Modell zu navigieren.

Datenarchitektur

Modelle des Typs *Datenarchitektur* stellen fachliche Konzepte und ihre Beziehungen dar. Der Modelltyp unterstützt Modellierer dabei, Informationen über den Einsatz der Anwendungslandschaft zu einander in Beziehung zu setzen.

Abbildung 5.4 zeigt einige wesentliche fachliche Konzepte einer Versicherung. So gehört beispielsweise zu jedem *Vertrag* ein *Antrag*. Es gibt verschiedene Arten von Verträgen (z.B. einen Lebensversicherungsvertrag, kurz *LV-Vertrag*). Ein Antrag wiederum ist eine Sammlung mehrerer Dokumente. Falls die Versicherung zwei getrennte Softwaresysteme einsetzt, um Anträge und Verträge zu beantworten, hilft

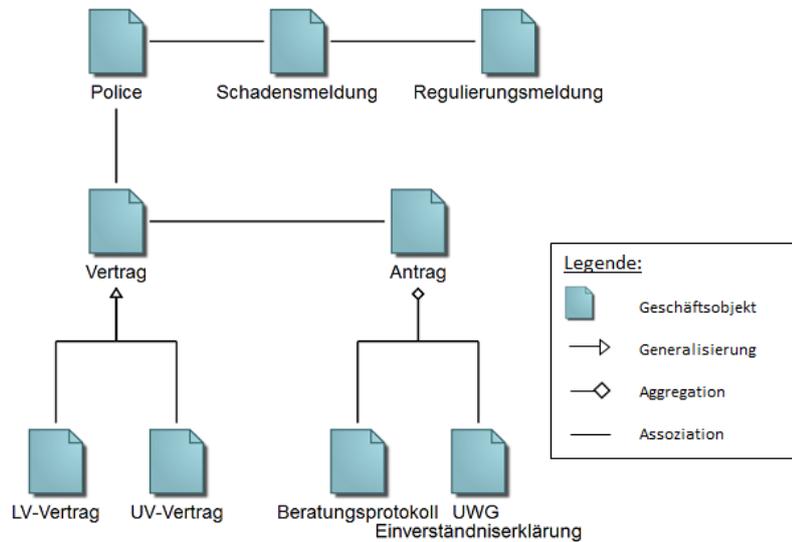


Abbildung 5.4: Beispiel für den Modelltyp Datenarchitektur

die in Abbildung 5.4 dargestellte Datenarchitektur, die fachlichen Zusammenhänge zwischen den beiden Softwaresystemen zu verstehen.

Anwendungsarchitektur

Modelle des Typs *Anwendungsarchitektur* stellen fachliche Architekturen dar. Anwendungsarchitekturen unterstützen unter anderem dabei, eine Anwendungslandschaft zu bewerten, den Ablauf von Umgestaltungsprojekten zu planen und die Veränderungen an die betroffenen Akteure zu kommunizieren.

Abbildung 5.5 zeigt einige zentrale Softwaresysteme in der Anwendungslandschaft einer Versicherung. Die *Bestandsverwaltung* stellt über Schnittstellen Daten zu Kunden und Verträgen bereit. Das *CRM* nutzt beide Schnittstellen. Das Softwaresystem zum Schadensmanagement greift hingegen nur auf die Vertragsdaten zu.

Das in Abbildung 5.3 dargestellte Kooperationszenario macht Schwächen in der IT-Unterstützung der Schadensbearbeitung deutlich. Ein Blick auf das Modell der Anwendungsarchitektur zeigt, dass es keine Schnittstellen zwischen dem CRM und dem Softwaresystem zum Schadensmanagement gibt. Daher ist die Anwendungslandschaft nicht in der Lage, den Geschäftsprozess „Schadensbearbeitung“ besser zu unterstützen.

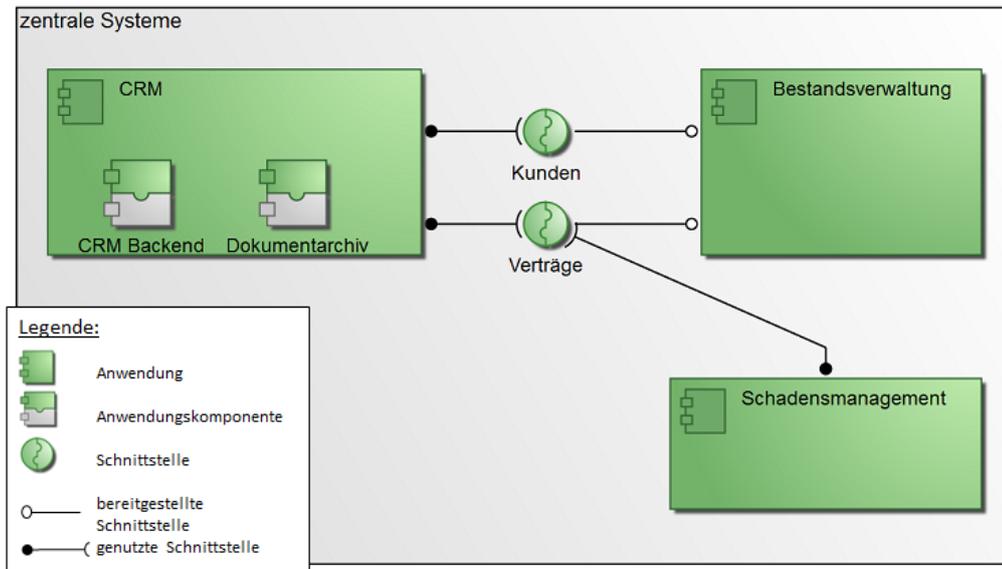


Abbildung 5.5: Beispiel für den Modelltyp Anwendungsarchitektur

Technologiearchitektur

Modelle des Typs *Technologiearchitektur* zeigen, welche technischen Abhängigkeiten zwischen Anwendungen bestehen. Der Modelltyp unterstützt dabei, Umgestaltungsprojekte z.B. durch Veränderung von Schnittstellen umzusetzen.

Abbildung 5.6 zeigt, wie das Softwaresystem *CRM* technisch aufgebaut ist. Es besteht aus zwei *Anwendungskomponenten*, die auf zwei Datenbanken zugreifen. Außerdem wird ein Datei-Server eingebunden. Anwendungskomponenten und Datenbanken sind auf zwei Server verteilt. Über eine REST-Schnittstelle² kann das CRM mit anderen Anwendungen kommunizieren.

²Representational State Transfer, siehe [TESW15]

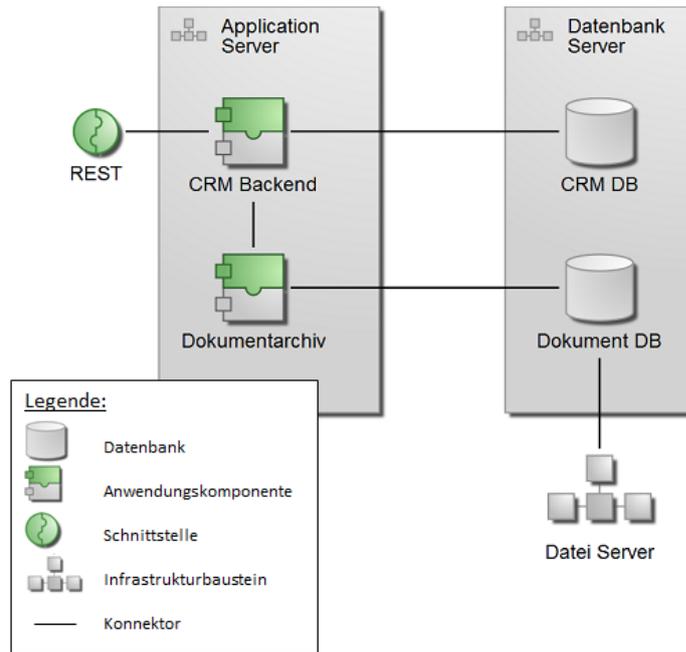


Abbildung 5.6: Beispiel für den Modelltyp Technologiearchitektur

5.3.3 Anwendung der Bausteine

Um die Bausteine auf ADOit anzuwenden, waren mehrere Arbeitsschritte erforderlich:

- Zunächst musste geprüft werden, welche Bausteine die in ADOit vorhandenen Modelltypen und Modellelementtypen bereits umsetzen.
- Für Umgestaltungsprojekte nicht geeignete Modelltypen und Modellelementtypen wurden identifiziert. Auf sie wird im Folgenden nicht eingegangen.
- Das Delta zwischen den vorhandenen und den benötigten Sprachmitteln wurde geschlossen, in dem ein Modelltyp und mehrere Modellelementtypen ergänzt wurden.
- Der Autor entschied, die nötigen Ergänzungen so vorzunehmen, dass die in ADOit existierenden Sprachmittel nicht angepasst werden müssen. Mit ADOit erstellte Modelle sind daher aufwärtskompatibel zum erweiterten ADOit.

Die folgenden Abschnitte beschreiben, wie die Bausteine angewendet wurden. Die resultierenden Modelltypen weisen einige syntaktische Besonderheiten auf, die nicht über das Metamodell des Modellierungswerkzeugs definiert werden konnten.

Diese Besonderheiten werden als Modellierungsregeln beschrieben. Im Anhang D werden die Modellierungsregeln formal spezifiziert, sodass eine werkzeuggestützte Regelprüfung implementiert werden kann. Im Rahmen des Anwendungsbeispiels wurde keine Regelprüfung in ADOit umgesetzt. Die Regeln gelten im Anwendungsbeispiel per Konvention.

Ob der beispielhafte Modellierungsansatz die in ihn gesetzten Erwartungen erfüllt, wird im Rahmen der Evaluation in Kapitel 6 betrachtet.

Modelltyp für Geschäftsprozesse (Baustein 1)

Zwar verfügt ADOit über einen Modelltyp zur Darstellung von Geschäftsprozessen, doch weist dieser weder eine anwendungsorientierte Modellierungssprache auf, noch eignet er sich zum Szenario-basierten Modellieren (Baustein 8). Daher wurde ein neuer Modelltyp *Kooperationsszenario* ergänzt, der diese beiden Bausteine umsetzt. Details zum Szenario-basierten Modellieren werden im Abschnitt zu Baustein 8 erläutert.

Als Vorbild für den neuen Modelltyp diente das „Kooperationsbild“ der eGPM (detaillierte Beschreibung siehe Anhang C). Kooperationsbilder werden in Workshops modelliert, ohne die Teilnehmer zuvor in der Modellierungssprache zu schulen. Daher veranschaulicht die eGPM, wie Baustein 1 so umgesetzt werden kann, dass alle Teilnehmer gleichberechtigt über Modelle kommunizieren können.

Für den neuen Modelltyp „Kooperationsszenario“ wurden die in ADOit vorhandenen Modellelementtypen „Akteur“ und „Anwendung“ wiederverwendet. Die weiteren Modellelementtypen und Relationstypen³ wurden vom Autor ergänzt und sind in der folgenden Auflistung fett gedruckt:

- Modellelementtypen: Akteur (AKT), Anwendung (ANW), **Arbeitsgegenstand** (ARB) und **referenziertes Szenario** (RSZ). Aktive Elemente eines Geschäftsprozesses sind vom Typ „Akteur“ oder „Anwendung“. Passive Elemente können von jedem Modellelementtyp sein.
- Relationstypen: Die Relationstypen werden als gerichtete Kanten visualisiert, die an einem aktiven Modellelement beginnen und zu einem passiven Modellelement hinführen.

³Um die Typisierung der Relationen übersichtlich zu halten, wird im Folgenden die Menge der Elemente eines Modellelementtyps durch den abgekürzten Namen des Modellelements ausgedrückt.

bearbeitet: Formal betrachtet gibt es zwei Relationstypen mit diesem Namen: $bearbeitet \subseteq AKT \times ARB$ und $bearbeitet \subseteq ANW \times ARB$. Beide drücken dasselbe Konzept aus – ein aktives Modellelement bearbeitet ein passives. ADOit ermöglicht, konzeptionell ähnliche Beziehungen in nur einem Relationstyp zu definieren. Der Autor nutzte dies, um die beiden Relationstypen zusammenzufassen.

initiiert: Analog zum Relationstyp „bearbeitet“ wurden die beiden formal zu unterscheidenden Typen $initiiert \subseteq AKT \times RSZ$ und $initiiert \subseteq ANW \times RSZ$ in einem Relationstyp zusammengefasst.

kommuniziert über: Auch hier können die aktiven Elemente von unterschiedlichem Typ sein, woraus sich zwei formal zu unterscheidende Relationstypen ergeben. Zusätzlich sind diese beiden Relationstypen dreistellig (aktives Element \times passives Element \times passives Element). Da ADOit nur zweistellige Relationstypen unterstützt, mussten die dreistelligen Relationstypen in zwei zweistellige Relationstypen aufgeteilt werden. Damit ergeben sich formal folgende Relationstypen, die in ADOit zu einem Relationstyp zusammengefasst wurden:

$$kommuniziert\ über \subseteq AKT \times ARB$$

$$kommuniziert\ über \subseteq ANW \times ARB$$

$$kommuniziert\ über \subseteq ARB \times AKT$$

$$kommuniziert\ über \subseteq ARB \times ANW$$

Wie die Modellelement- und Relationstypen des Kooperationsszenarios eingesetzt werden, beschreiben Modellierungsregeln (siehe Anhang D). Die wichtigsten Modellierungsregeln sind:

- Eine Regel für den dreistelligen Relationstyp „kommuniziert über“. Sie beschreibt, wie eine dreistellige Relation durch zwei semantisch zusammengehörigen Relationen ausgedrückt wird. Auf Grund dieser Regel dürfen „kommuniziert über“-Relationen immer nur paarweise modelliert werden.
- Die Abfolge von Tätigkeiten in einem Arbeitsablauf wird durch nummerierte Arbeitsgegenstände ausgedrückt. Die Abfolge der Nummern muss eine monoton steigende Folge von natürlichen Zahlen sein, beginnend mit Eins.

Die mehrfache Vergabe einer Nummer ist erlaubt und drückt aus, dass die Tätigkeiten gleichzeitig durchgeführt werden.

- Um die Kooperation zu veranschaulichen, darf jeder Akteur und jede Anwendung nur einmal in einem Modell enthalten sein. Noch deutlicher wird die Kooperation der Akteure und Anwendungen dadurch, dass jeder Arbeitsgegenstand nur in einer einzigen Relation verwendet werden darf.

In Kooperationsszenarien können Geschäftsprozesse mit ihren anwendungsorientierten Aspekten dargestellt werden. Tabelle 5.3 zeigt, wie sich solche Aspekte mit den Modellelementtypen des Kooperationsszenarios ausdrücken lassen.

Modelltyp für fachliche Architekturen (Baustein 2)

Der Modelltyp *Anwendungsarchitektur*⁴ kann fachliche Architekturen darstellen, wenn man die Modellelemente mit fachlichen Bezeichnungen benennt.

In der Anwendungsarchitektur stehen Modellelementtypen zur Verfügung, die nicht für die Umsetzung dieses Bausteins benötigt werden. Im Folgenden werden nur die relevanten Modellelementtypen aufgeführt:

- Modellelementtypen: Anwendung (ANW), Anwendungskomponente (AWK), Service (SRV), Schnittstelle (SST)
- Relationstypen:

bereitgestellte Schnittstelle \subseteq ANW \times SST

bereitgestellter Service \subseteq ANW \times SRV

genutzte Schnittstelle \subseteq ANW \times SST

genutzter Service \subseteq ANW \times SRV

Informationsfluss \subseteq ANW \times ANW

zugehörige Anwendungskomponente \subseteq ANW \times AWK

⁴Im ADOit Modellierungswerkzeug heißt der Modelltyp *Informationssystemarchitektur*. „Anwendungsarchitektur“ ist eine Darstellungsoption, in der nur eine Teilmenge der Modellelemente des Modelltyps verfügbar sind. Damit realisiert die „Anwendungsarchitektur“ einen Subtyp der „Informationssystemarchitektur“.

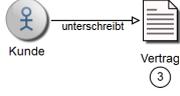
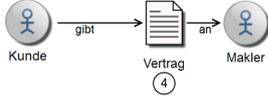
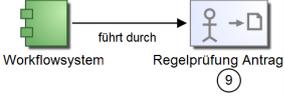
Vorgabe aus Baustein 1	Umsetzung in ADOit	Beispiel
Akteur	Modellelementtyp <i>Akteur</i>	 Kunde
Arbeitsgegenstand und -ergebnis	Modellelementtyp <i>Arbeitsgegenstand</i>	 Vertrag
Geschäftsprozess	Modellelementtyp <i>referenziertes Szenario</i>	 Nachbearbeitung
Softwaresystem	Modellelementtyp <i>Anwendung</i>	 Bestandsverwaltung
Werkzeug	<i>Arbeitsgegenstand</i> mit IT-Unterstützung durch eine verknüpfte <i>Anwendung</i>	 Vorgangsmappe <Workflowsystem>
Tätigkeit	<p>Relationstyp <i>bearbeitet</i>: Akteur oder Anwendung bearbeitet Arbeitsgegenstand</p> <p>Relationstyp <i>kommuniziert über</i>: Akteur oder Anwendung kommuniziert über Arbeitsgegenstand mit Akteur oder Anwendung</p> <p>Relationstyp <i>initiiert</i>: Akteur oder Anwendung initiiert referenziertes Szenario</p>	  
Aufgabe	Zusammenhängende Abfolge von <i>Tätigkeiten</i> . Die Abfolge ergibt sich aus der Nummerierung der <i>Tätigkeiten</i> .	
Arbeitsablauf	Abfolge von <i>Aufgaben</i> . Die Abfolge ergibt sich aus der Nummerierung der <i>Tätigkeiten</i> .	

Tabelle 5.3: Anwendungsorientierte Modellierungssprache des Kooperationszenarios

Die zwischen den Anwendungen ausgetauschten Geschäftsobjekte werden nicht dargestellt. An den verwendeten Services und Schnittstellen können jedoch – im Modell nicht sichtbare – Referenzen auf Geschäftsobjekte einer *Datenarchitektur* (s.u.) dokumentiert werden. Um sichtbar zu machen, wie Anwendungen fachlich zusammenhängen, sind Schnittstellen und Services in Anwendungsarchitekturen mit fachlichen Bezeichnungen zu benennen. Technische Begriffe, welche die eingesetzte Schnittstellentechnik beschreiben, sind zu vermeiden. Diese Modellierungsregel kann nicht formal spezifiziert werden.

Der zentrale Modellelementtyp einer fachlichen Architektur in ADOit ist die „Anwendung“. Daher gilt die Modellierungsregel, dass jede Anwendung nur einmal pro Modell abgebildet werden darf.

Modelltyp für fachliche Konzepte (Baustein 3)

Der Modelltyp *Datenarchitektur*⁵ eignet sich, um fachliche Konzepte und ihre Beziehungen abzubilden.

In der Datenarchitektur stehen Modellelementtypen zur Verfügung, die nicht für die Umsetzung dieses Bausteins benötigt werden. Im Folgenden werden nur die relevanten Modellelementtypen aufgeführt:

- Modellelementtypen: Geschäftsobjekt (GOB)
- Relationstypen:

Aggregation \subseteq GOB \times GOB

Assoziation \subseteq GOB \times GOB

Generalisierung \subseteq GOB \times GOB

Komposition \subseteq GOB \times GOB

Damit Datenarchitekturen als Glossar dienen können, darf jedes Geschäftsobjekt nur einmal pro Modell dargestellt werden.

⁵„Datenarchitektur“ ist eine weitere Darstellungsoption des ADOit-Modelltyps „Informationssystemarchitektur“.



Abbildung 5.7: Arbeitsgegenstand ohne (links) und mit (rechts) offenen Fragen

Modelltyp für Bausteinsichten (Baustein 4)

Der Modelltyp *Technologiearchitektur* eignet sich, um den inneren Aufbau von Anwendungen und Abhängigkeiten zwischen Anwendungen darzustellen.

In der Datenarchitektur stehen Modellelementtypen zur Verfügung, die nicht für die Umsetzung dieses Bausteins benötigt werden. Im Folgenden werden nur die relevanten Modellelementtypen aufgeführt:

- Modellelementtypen: Anwendungskomponente, Datenbank, Infrastrukturbau- stein, Schnittstelle
- Relationstypen: Konnektor (verbindet zwei Elemente beliebiger Typen)

Dieser Modelltyp muss flexibel genug sein, um beispielsweise verteilte, virtuelle und redundante Infrastruktur darstellen zu können. Um die Flexibilität zu wahren, werden keine Modellierungsregeln für Technologiearchitekturen spezifiziert.

Vage Informationen grafisch kennzeichnen (Baustein 5)

Analog zum Kooperationsbild (siehe Anhang C) hat der Autor ein textuelles Attribut „offene Fragen“ in den Modellelementtyp „Arbeitsgegenstand“ und den Modelltyp „Kooperationsszenario“ integriert. Offene Fragen werden im Modell als Icon visualisiert (siehe Abbildung 5.7) und über einen Texteditor eingegeben (siehe Abbildung 5.8).

Die anderen, aus ADOit übernommenen Modellelemente und Modelltypen wurden nicht um das Attribut „offene Fragen“ erweitert, damit die Modellierungssprache von ADOit aufwärtskompatibel zu den Erweiterungen bleibt. Diese Entwurfsent- scheidung wurde am Anfang dieses Kapitels erläutert.

Modelle verknüpfen (Baustein 6)

ADOit bietet mehrere Möglichkeiten, um Modelle miteinander zu verknüpfen, die dasselbe Original in verschiedenen Sichten darstellen. Zum einen verfügt ADOit

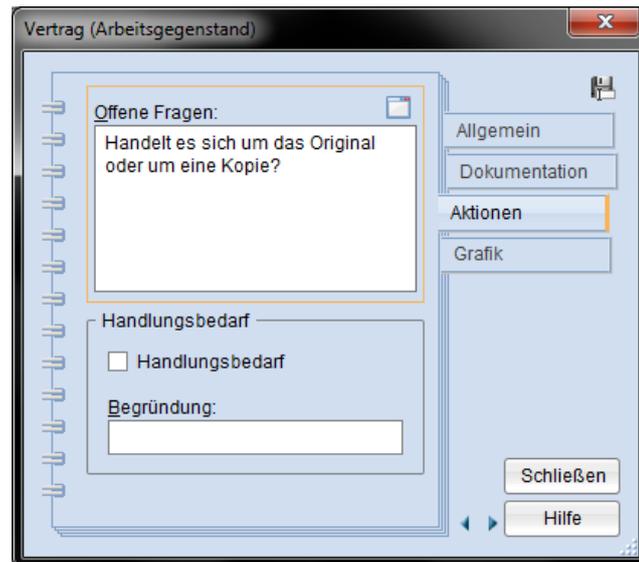


Abbildung 5.8: Vage Information als „offene Frage“ dokumentieren

über ein Datenbank-basiertes Informationsmodell, in dem ein Modellelement wie z.B. der Akteur „Makler“ nur einmal existiert – unabhängig davon, in wie vielen Modellen dieses Modellelement grafisch dargestellt wird (siehe Abschnitt 5.3.1).

Zum anderen können Modellelemente über Referenzen mit anderen Modellelementen verknüpft werden. Zwei Beispiele sind die Referenzen *verknüpfte Geschäftsobjekte* \subseteq ARB \times GOB und *verknüpfte Anwendungen* \subseteq ARB \times ANW. Abbildung 5.9 zeigt, wie ein Arbeitsgegenstand eines Kooperationsszenarios mit dem Geschäftsobjekt „Vorgangsmappe“ einer Datenarchitektur in Beziehung gesetzt wurde. Der Arbeitsgegenstand ist zudem mit der Anwendung „Workflowsystem“ verknüpft.

Zustand kennzeichnen (Baustein 7)

In den Modelltypen „Technologiearchitektur“, „Anwendungsarchitektur“ und „Datenarchitektur“ können Modellelemente mit den zeitlichen Attributen „gültig von“ und „gültig bis“ versehen werden. Dadurch ist es möglich, in einem Modell unterschiedliche Zustände der modellierten Anwendungslandschaft darzustellen. Beispielsweise zeigt Abbildung 5.10 (oben) Anwendungen, die zum Zeitpunkt der Modellierung schon in Betrieb waren und Anwendungen, die erst später in Betrieb genommen werden. Durch die Werkzeugfunktion „Zeitfilter“ kann der gerade im Modell dargestellte Zeitpunkt eingestellt werden. Abbildung 5.10 (Mitte) zeigt den Zustand bis zum 31.12.2015 und Abbildung 5.10 (unten) den Zustand ab 01.01.2016.

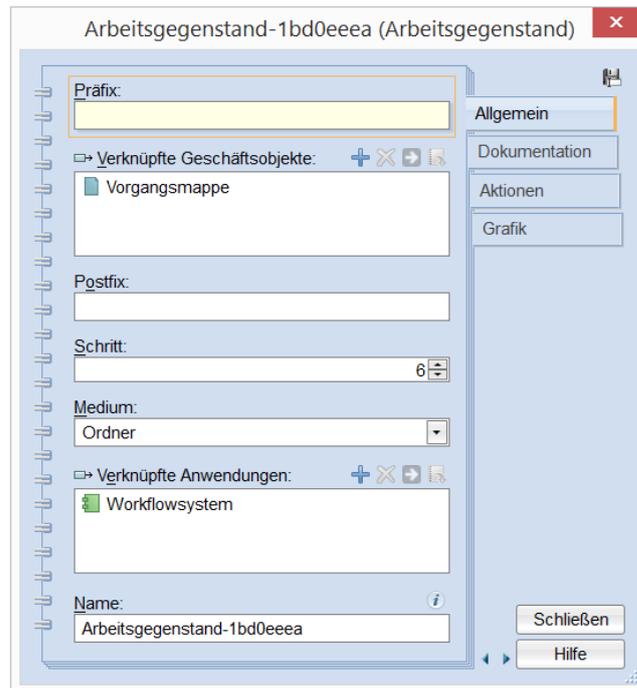


Abbildung 5.9: Modellelemente über Referenzen miteinander verknüpfen

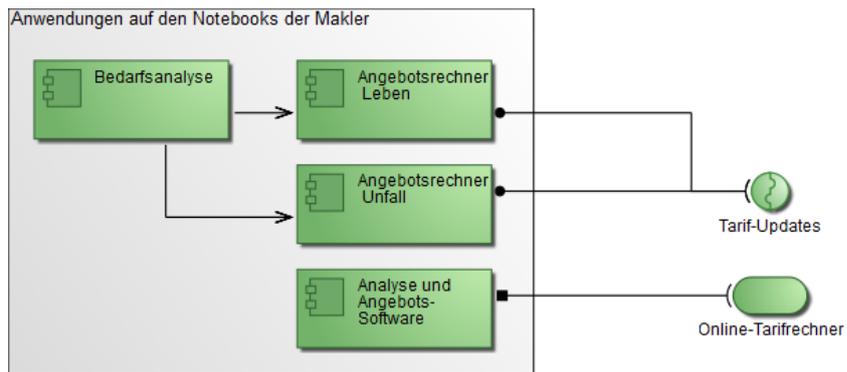
In diesem Beispiel werden also mit Jahreswechsel einige Anwendungen abgeschaltet und durch eine neue Anwendung ersetzt.

Für Kooperationsszenarien mit ihren nummerierten Tätigkeiten ist der Ansatz mit Zeit-Attributen und Zeitfilter hingegen keine gute Option. Zum einen ließe sich die Reihenfolge der Tätigkeiten schwer nachvollziehen, wenn durch einen Zeitfilter einige Tätigkeiten aus- oder eingeblendet würden. Zum anderen wäre dieser Ansatz nicht Szenario-basiert (siehe Abschnitt zu Baustein 8). Statt alternative Zustände eines Arbeitsablaufs in einem Modell unterzubringen, wird daher für jeden Zustand ein eigenes Kooperationsszenario erstellt und mit einem Attribut *Status* als Ist- oder Soll-Modell gekennzeichnet.

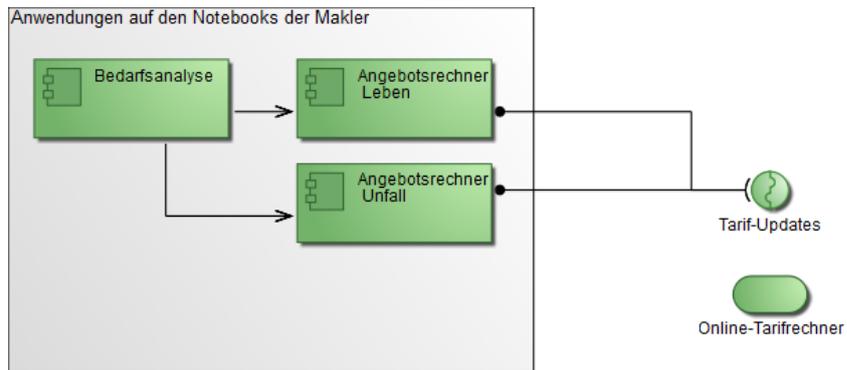
ADOit unterstützt den paarweisen Vergleich von Modellen durch eine tabellarische Auflistung der Unterschiede (siehe Abbildung 5.11). Mit dieser Funktion können Unterschiede zwischen Ist- und Soll-Zuständen werkzeuggestützt analysiert werden.

Umsetzung von Baustein 8 – Szenario-basiertes Modellieren

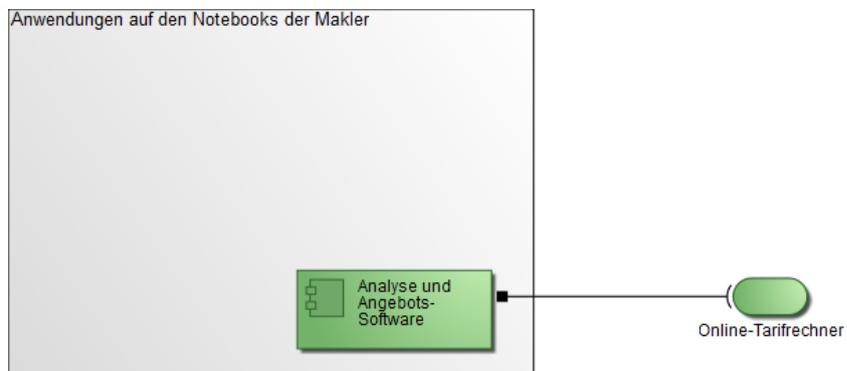
Das Kooperationsszenario unterstützt Szenario-basiertes Modellieren:



(a) Zeitfilter deaktiviert



(b) Zeitfilter aktiv – dargestellter Zeitpunkt: 31.12.2015



(c) Zeitfilter aktiv – dargestellter Zeitpunkt: 01.01.2016

Abbildung 5.10: Anwendungsarchitektur mit Zeitfilter

Klasse / Beziehung	In beiden Modellen	Nur im Modell Ausstellung ei...	Nur im Modell Ausstellun...
Arbeitsgegenstand			
	Arbeitsgegenst...	Arbeitsgegenstand-319c8...	
	Arbeitsgegenst...		
	Arbeitsgegenst...		

Abbildung 5.11: Modellvergleich zwischen einem Ist- und einem Soll-Kooperationsszenario

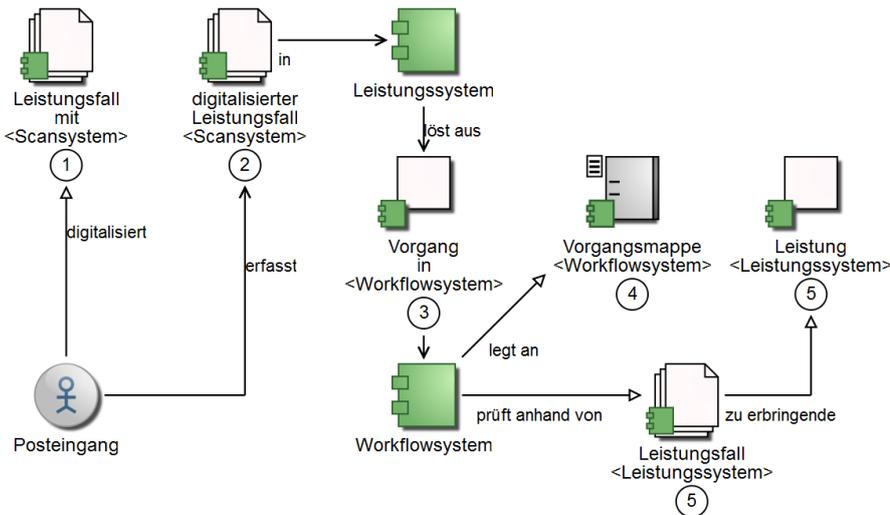


Abbildung 5.12: Sequenz von Tätigkeiten (Ausschnitt aus einem Kooperationsszenario)

- Im Modellattribut „Beschreibung“ kann erläutert werden, welche Kriterien das Szenario von anderen Prozessverläufen abgrenzen.
- Da keine Fallunterscheidungen dargestellt werden können, zwingt die Modellierungssprache den Modellierer, sich auf ein Szenario festzulegen.
- Die Abfolge von Tätigkeiten, welche den Arbeitsablauf ausmacht, wird durch Nummerierung dargestellt (siehe Abbildung 5.12).

Szenario-basiertes Modellieren benötigt ein entsprechendes Vorgehen. Da der Modelltyp „Kooperationsbild“ der eGPM (detaillierte Beschreibung siehe Anhang C) als Vorbild für das Kooperationsszenario diente, lässt sich das Modellierungsvorgehen übertragen. In der eGPM leiten die Fragen *Wer macht was mit welchem Arbeitsgegenstand?* und *Wer kommuniziert über welche Arbeitsgegenstände mit wem?* die Modellierung:

„Von diesen Grundfragen angetrieben entstehen sehr konkrete Prozessdarstellungen, die explizit die Kooperation und Koordination zwischen den beteiligten Akteuren betrachten. Darüber hinaus wird klar herausgearbeitet, welche [Tätigkeiten] auf welche Art durch [Softwaresysteme] unterstützt werden.“ [BH12, S.8]

Dies kann auf den beispielhaften Ansatz übertragen werden:

- Welche Akteure und Anwendungen bearbeiten welche Arbeitsgegenstände?

- Welche Akteure und Anwendungen kommunizieren über welche Arbeitsgegenstände mit welchen anderen Akteuren und Anwendungen?

Gemeinschaftliches Modellieren (Baustein 9)

Dieser Baustein wurde aus der eGPM entnommen. Auf Grund der Ähnlichkeit des Modelltyps „Kooperationsszenario“ mit dem eGPM-Modelltyp „Kooperationsbild“ kann, wie in Baustein 9 beschrieben, mit Kooperationsszenarien gemeinschaftlich modelliert werden: Sie eignen sich, um Workshops mit Fach- und IT-Experten abzuhalten, in denen Arbeitsabläufe modelliert werden. Im großflächig projizierten Modellierungswerkzeug entstehen Kooperationsszenarien vor den Augen aller Workshop-Teilnehmer und können unmittelbar validiert werden.

Ob gemeinschaftliches Modellieren auch auf die anderen Modelltypen des erweiterten ADOit anwendbar ist, wurde im Rahmen dieser Arbeit nicht untersucht.

Zweck beschreiben (Baustein 10)

Um diesen Baustein umzusetzen, musste beschrieben werden, welche Rollen bei welchen Umgestaltungstätigkeiten durch den Modellierungsansatz unterstützt werden. Für das Kooperationsszenario laut die Beschreibung folgendermaßen:

Ein Kooperationsszenario stellt einen exemplarischen Arbeitsablauf dar. Es zeigt, wie menschliche Akteure miteinander und mit Anwendungen kooperieren und Arbeitsgegenstände bearbeiten.

Einsatzmöglichkeiten:

- Für Modellierer: Informationen über die Anwendungslandschaft erfassen
- Für Projektleitung, Architekten: Ist- und Soll-Zustände bewerten, Veränderungen an Projektmitarbeiter und Linienorganisation kommunizieren
- Für Software-Entwickler, Tester: veränderte Abläufe verstehen

Im Anhang D findet sich für jeden Modelltyp eine Abbildung, die den Modelltyp erklärt und seinen Zweck beschreibt. Die Abbildungen umfassen jeweils:

- Eine Beschreibung des Einsatzzwecks – analog zum oben geschilderten Beispiel: Wer (welche Rolle) soll durch den Modelltyp bei welchen Umgestaltungstätigkeiten unterstützt werden?
- Eine Legende der Modellelementtypen und ihrer Relationen.
- Eine Legende der im Modell nicht sichtbaren, nur über das Modellierungswerkzeug editierbaren Relationen.
- Ein Beispielmodell.

5.4 Zusammenfassung

Am Anfang dieses Kapitels wurde eine Zielvorstellung für einen „idealen“ Modellierungsansatz für Umgestaltungsprojekte skizziert. Realisieren lässt sich ein solches Ideal jedoch nicht, da es nicht auf konkrete, branchen- und unternehmensspezifische Besonderheiten eingeht. Daher präsentiert dieses Kapitel zehn Bausteine, welche die für Umgestaltungsprojekte Verantwortlichen in die Lage versetzen, existierende Modellierungsansätze an ihre Umgestaltungsprojekte anzupassen und so dem Ideal näher zu kommen. Die Bausteine wurden aus den in Kapitel 4 bewerteten Ansätzen abgeleitet und beantworten die Forschungsfrage, welche Mittel ein Modellierungsansatz aufweisen muss, damit er sich besonders gut für Umgestaltungsprojekte eignet.

Da die Bausteine keine konkrete Modellierungssprache und keine konkreten Vorgehensweisen beschreiben, bleiben sie abstrakt. Um ihre Anwendbarkeit zu zeigen und die an den Bausteinen Interessierten bei der Adaptierung von Ansätzen zu unterstützen, wurde ein beispielhafter Modellierungsansatz entwickelt. Dazu wurden die Bausteine auf den kommerziellen, werkzeuggestützten Modellierungsansatz ADOit angewendet.

Ob mit Hilfe der Bausteine nützliche Modellierungsansätze entwickelt werden können, wird im nächsten Kapitel überprüft.

Kapitel 6

Evaluation der Bausteine

Ziel dieses Kapitels ist, die Nützlichkeit der Bausteine zu belegen. Dazu werden die in der Einleitung dieser Arbeit eingeführten Qualitätsmerkmale wieder aufgegriffen. Anhand mehrerer Indikatoren wird belegt, dass die gesteckten Ziele erreicht wurden.

6.1 Auswahl von Qualitätsmerkmalen

Die Qualität komplexer Lösungen – wie Softwaresysteme und Modelle – lässt sich nicht einfach objektiv beurteilen. Zu viele verschiedene Aspekte einer Lösung, von ihrer Gebrauchstauglichkeit bis hin zur Güte ihrer „handwerklichen“ Umsetzung sind potentiell von Bedeutung. In der Softwaretechnik und in der konzeptuellen Modellierung werden daher *Qualitätsmerkmale* eingesetzt, um Softwaresysteme und Modelle anhand ausgewählter Aspekte qualitativ einzuschätzen. Qualitätsmerkmale können auf mehrere Teilmerkmale heruntergebrochen werden. Ob ein (Teil-)Merkmal erfüllt ist, wird entweder durch Maßzahlen gemessen oder anhand von Indikatoren eingeschätzt. Ein Beispiel hierfür ist der ISO/IEC-Standard 25000 („Software Quality Requirements and Evaluation“, siehe [ISO14]):

- Einige der Qualitätsmerkmale, wie z.B. der Ressourcenverbrauch, können gemessen werden.
- Andere, wie die Änderbarkeit eines Softwaresystem, ist anhand von Indikatoren einschätzbar.

Gleiches gilt für die Qualitätsmerkmale, die Mohagheghi et al. in [MDN08] für Modelle aufstellen: Beispielsweise kann die syntaktische Korrektheit objektiv

überprüft werden. Hingegen sind die Verständlichkeit und die Änderbarkeit von Modellen nur heuristisch einschätzbar.

Die Wahl der Qualitätsmerkmale für die in dieser Arbeit beschriebenen Forschungsergebnisse ist eng mit der eingesetzten Forschungsmethodik verknüpft. Wie in Abschnitt 1.3 beschrieben wurde, liegt dieser Arbeit die Forschungsmethodik *Design Research* zugrunde. Mit Design Research erzielte Ergebnisse sollen die Praxis verbessern [MS95, S.251]. Mit den Bausteinen und dem beispielhaften Modellierungsansatz wurden Ergebnisse in Form von Konzepten und Ausprägungen erarbeitet. Um zu evaluieren, ob mit diesen die angestrebte Verbesserung der Praxis erreicht werden konnte, wird dieses übergeordnete Qualitätsmerkmal auf drei Teilmerkmale heruntergebrochen: Relevanz, Anwendbarkeit und Praxistauglichkeit. Da diese Arbeit keine Methode zur Entwicklung von Modellierungsansätzen – also kein Artefakt vom Typ „Methode“ – zum Ziel hatte, wird nicht evaluiert, ob die Bausteine für Methoden- und Werkzeugentwickler effizient anzuwenden sind.

Durch Indikatoren kann abgeschätzt werden, ob die Bausteine die Qualitätsmerkmale Relevanz, Anwendbarkeit und Praxistauglichkeit aufweisen:

Relevanz: Die Bausteine müssen die Modellierungsprobleme minimieren.

Ausgangspunkt für die Bausteine waren sechs Modellierungsprobleme, die bei einer empirischen Untersuchung von Umgestaltungsprojekten festgestellt wurden (siehe Abschnitt 3.3). Ob die Bausteine helfen, die Probleme bei der Modellierung zu minimieren, kann durch folgende Indikatoren bewertet werden:

- Gibt es für jedes Modellierungsproblem einen Baustein, der das Problem minimieren kann?
- Erfüllt ein mit den Bausteinen angepasster Modellierungsansatz die Anforderungen aus dem Anforderungskatalog?

Anwendbarkeit: Mit den Bausteinen muss ein Modellierungsansatz angepasst werden können.

Ob die Bausteine dazu geeignet sind, kann durch einen *Proof-of-Concept* demonstriert werden:

- Lässt sich ein Beispiel konstruieren, das die Anwendung der Bausteine zeigt?

Praxistauglichkeit: Ein mit den Bausteinen angepasster Ansatz muss für den Unternehmenseinsatz geeignet sein.

Da die Bausteine Probleme aus der Praxis aufgreifen, muss die Lösung der Probleme praxistauglich sein:

- Können die Bausteine auf einen werkzeuggestützten Ansatz angewendet werden, der sich im Unternehmenseinsatz bewährt hat?
- Kann der Proof-of-Concept eingesetzt werden, um Fragestellungen aus einem echten Umgestaltungsprojekt durch Modelle zu beantworten? Dazu wird ein in [Ros09] beschriebenes Projekt herangezogen, das in dieser Arbeit mehrfach als Beispiel verwendet wurde (siehe Abschnitt 3.5 und Anhang B).

6.2 Bewertung der Qualitätsmerkmale

In diesem Abschnitt wird anhand der Indikatoren bewertet, ob die Bausteine die geforderten Qualitätsmerkmale aufweisen. Einige Ausprägungen beziehen sich auf den Modellierungsansatz, der als Anwendungsbeispiel entwickelt wurde. Das heißt, dass die Qualität des Anwendungsbeispiels die Evaluationsergebnisse der Bausteine beeinflusst.

Die Indikatoren werden bis auf eine Ausnahme durch den Autor argumentativ bewertet. Die Ausnahme bildet ein Indikator für die Praxistauglichkeit. Dieser wurde mittels eines Fragebogens von einem Experten empirisch bewertet.

6.2.1 Anwendbarkeit

Dass die Bausteine anwendbar sind, zeigt der als Anwendungsbeispiel entwickelte Ansatz (siehe Abschnitt 5.3). Dieser Proof-of-Concept demonstriert, wie man mit Hilfe der Bausteine einen bestehenden Ansatz adaptiert:

Der als Grundlage gewählte Ansatz eignet sich bereits, um Anwendungslandschaften darzustellen und weist vier von zehn Bausteinen auf. Ein weiterer Baustein, Kennzeichnung als Ist- oder Soll-Modell (Baustein 7), war teilweise vorhanden und wurde adaptiert. Die fehlenden fünf Bausteine wurden vom Autor ergänzt:

- Ein Modelltyp zur Darstellung von Geschäftsprozessen (Baustein 1).

- Vage Informationen im Modell kennzeichnen (Baustein 5).
- Szenario-basiertes Modellieren von Geschäftsprozessen (Baustein 8).
- Beteiligte aus unterschiedlichen Aufgabengebieten modellieren gemeinsam unter Anleitung eines Moderators (Baustein 9).
- Beschreibung, welcher Zweck mit welchen Modelltypen erreicht werden kann (Baustein 10).

Zusammenfassend wurde ein praxisrelevanter, werkzeuggestützter, kommerzieller Modellierungsansatz gegen die Liste der Bausteine abgeglichen und der Ansatz um die Hälfte der Bausteine ergänzt. Daher stellt der Ansatz ein realistisches Anwendungsbeispiel für den Zweck der Bausteine dar.

6.2.2 Relevanz

Wie relevant die Bausteine sind, wird anhand von zwei Indikatoren bewertet:

1. Indikator: Minimierung der Modellierungsprobleme

Die in Abschnitt 3.3 eingeführten Modellierungsprobleme werden hier kurz wiederholt und dann begründet, welche Bausteine helfen, die Probleme zu minimieren:

Modellierungsproblem 1: Modelle müssen darstellen können, *wie* die Anwendungslandschaft die Geschäftsprozesse unterstützt. Es reicht nicht aus, zu zeigen, *dass* ein Softwaresystem einen Prozess unterstützt.

Baustein 1 beschreibt eine anwendungsorientierte Modellierungssprache für Geschäftsprozesse. Mit dieser kann im Geschäftsprozessmodell sichtbar gemacht werden, wie IT-gestützte Werkzeuge eingesetzt und digitale Arbeitsgegenstände und -ergebnisse bearbeitet werden.

Modellierungsproblem 2: Die zu modellierende Informationsmenge ist schlecht handhabbar.

In dieser Arbeit wurde dargelegt, dass Anwendungslandschaften in der Praxis nicht vollständig in Modelle abgebildet werden können – egal, wie viel „mehr“ und „genauer“ man modelliert. Unvollständigkeit wird daher als gegeben akzeptiert und in Form von Szenario-basiertem Modellieren als Konzept

eingesetzt, um die Informationsmenge handhabbar zu machen. Die an der Modellierung beteiligten Akteure legen gemeinschaftlich Kriterien fest, nach denen die zu modellierenden Szenarien ausgewählt werden (Baustein 8).

Modellelemente in einem Informationsmodell zu speichern und in mehreren grafischen Modellen wiederzuverwenden, stellt eine weitere Möglichkeit dar, die zu modellierende Informationsmenge einzugrenzen. Eine Voraussetzung dafür schaffen die Bausteine 1–4 und 6. Diese beschreiben mehrere Modelltypen und zeigen, wie Modelle dieser Typen durch wiederverwendbare Modellelemente eine integrierte Sicht auf eine Anwendungslandschaft darstellen.

Modellierungsproblem 3: Bei der Modellierung können widersprüchliche Informationen auftreten.

Ein Baustein beschreibt, wie Widersprüche gekennzeichnet werden können (Baustein 5). Der gemeinschaftliche Modellierungsprozess (Baustein 9) hilft dabei, Widersprüche festzustellen.

Widersprüche sind ein Symptom dafür, dass es in der konzeptuellen Modellierung keine objektiv richtigen Modelle gibt – wie Box und Draper prägnant zusammenfassen: „all models are wrong“ [BD87, S.74]. Szenario-basiertes Modellieren (Baustein 8) erleichtert, die Brauchbarkeit von Modellen einzuschätzen.

Modellierungsproblem 4: Abhängigkeiten und Einsatz von Softwaresystemen sind besonders schwierig zu modellieren, wenn die Softwaresysteme abteilungs- und anwendungsübergreifende Geschäftsprozesse unterstützen.

Baustein 1 (Modelltyp zur Darstellung von Geschäftsprozessen) ermöglicht, den Blickwinkel der in der Anwendungslandschaft arbeitenden Menschen darzustellen. Beteiligt man Vertreter verschiedener Abteilungen am gemeinschaftlichen Modellierungsprozess (Baustein 9), kann modelliert werden, wie abteilungs- und anwendungsübergreifende Geschäftsprozesse durch Softwaresysteme unterstützt werden.

Modellierungsproblem 5: Abhängigkeiten zwischen Softwaresystemen sind schwer abzubilden, wenn sie sich nicht aus technischen Schnittstellen ergeben.

Ein gemeinschaftlicher Modellierungsprozess Baustein 9 hilft dabei, solche Abhängigkeiten zu identifizieren. Der in Baustein 1 beschriebene Modelltyp mit seiner anwendungsorientierten Modellierungssprache eignet sich, die Abhängigkeiten nachvollziehbar zu modellieren.

Modellierungsproblem 6: Veränderungen einer Anwendungslandschaft über die Zeit müssen dargestellt werden.

Baustein 7 ermöglicht, Modelle als Ist- oder Soll-Modell zu kennzeichnen und damit den Zustand der dargestellten Anwendungslandschaft auszudrücken.

2. Indikator: Bewertung anhand des Anforderungskatalogs

Als zweiter Indikator für die Relevanz der Bausteine dient der Anforderungskatalog aus Abschnitt 3.5. Tabelle 6.1 zeigt, dass das Anwendungsbeispiel alle Anforderungen voll oder zumindest teilweise erfüllt. Die Bewertung erfolgte nach dem in Kapitel 4 vorgestellten Schema: + (Anforderung erfüllt), •(Anforderung teilweise erfüllt) und – (Anforderung nicht erfüllt).

Anforderung 1 (IT-Unterstützung)	Bewertung: +
---	--------------

Arbeitsgegenstände und -ergebnisse werden als „Arbeitsgegenstand“ modelliert und über ein Attribut „verknüpfte Anwendung“ mit Anwendungen einer Anwendungsarchitektur verknüpft. Ein Icon am Arbeitsgegenstand visualisiert die Verknüpfung.

Anforderung 2 (Manuelle und IT-gestützte Tätigkeiten)	Bewertung: +
--	--------------

Kooperationsszenarien zeigen, wie menschliche und IT-basierte Akteure Tätigkeiten verrichten: Die Akteure bearbeiten Arbeitsgegenstände, setzen sie als Kommunikationsmittel ein oder geben sie weiter. Manuelle und IT-gestützte Tätigkeiten können anhand der IT-Unterstützung der eingesetzten Arbeitsgegenstände unterschieden werden.

Anforderung 3 (Koordination)	Bewertung: +
-------------------------------------	--------------

Kooperationsszenarien machen durch die Relation „kommuniziert (über)“ explizit, wie Akteure ihre Arbeitsabläufe untereinander koordinieren.

Anforderung 4 (Fachliche Konzepte)	Bewertung: +
---	--------------

Mit dem Modelltyp „Datenarchitektur“ können fachliche Konzepte in Beziehung gesetzt werden.

Anforderung 5 (Fachliche Architektur)	Bewertung: +
Mit dem Modelltyp „Anwendungsarchitektur“ können fachliche Architekturen dargestellt werden.	
Anforderung 6 (Technologische Sichten)	Bewertung: +
Mit dem Modelltyp „Technologiearchitektur“ können technologische Sichten (Bausteinsichten) dargestellt werden.	
Anforderung 7 (Beziehung zwischen Sichten)	Bewertung: +
Beziehungen zwischen Sichten sind auf verschiedenen Wegen realisierbar: Ein im Informationsmodell enthaltenes Modellelement kann in mehreren Modellen dargestellt werden. Modellelemente können über Attribute mit Modellelementen aus anderen Modellen verknüpft werden.	
Anforderung 8 (Zweifel an Brauchbarkeit)	Bewertung: ●
Der Modellelementtyp „Arbeitsgegenstand“ des Kooperations szenarios verfügt über ein auswertbares, textuelles Attribut „offene Fragen“. Ist dieses Attribut an einem Modellelement gesetzt, wird dessen Darstellung um ein ?-Symbol ergänzt. Andere Modellelementtypen weisen dieses Attribut nicht auf.	
Anforderung 9 (Validierung im Modellierungsprozess)	Bewertung: +
In Workshops unter Leitung eines Modellierers und Moderators modellieren die betroffenen Akteure gemeinschaftlich und können unmittelbar Rückmeldung geben.	
Anforderung 10 (Vages Modellieren)	Bewertung: +
Kooperations szenarien stellen exemplarische Prozessverläufe dar. Geschäftsprozesse werden nicht „vollständig“ modelliert, sondern nur die für den Modellzweck relevante Varianten. Dazu müssen bewusste Annahmen getroffen werden.	
Anforderung 11 (Einsatzszenarien)	Bewertung: ●
Der Zweck der Modelltypen – welche Umgestaltungstätigkeit sie unterstützen sollen – wird grob umrissen.	
Anforderung 12 (Modellmanipulation und Analyse)	Bewertung: ●
Modellmanipulation und -analyse werden nicht explizit beschrieben, aber durch das Werkzeug unterstützt: Mit einem „Zeitfilter“ können Ist- und Soll-Zustände von Anwendungslandschaften analysiert werden, das Werkzeug kann Modelle paarweise miteinander vergleichen und es ermöglicht Abfragen auf das Informationsmodell.	

Anforderung 13 (Status der Modelle)

Bewertung: ●

Kooperationsszenarien können als Ist- und Soll-Modelle mit gegebenenfalls noch vorhandenen offenen Fragen gekennzeichnet werden. In den übrigen Modelltypen werden Soll-Zustände auf Ebene von Modellelementen als „Lebenszyklus“ dokumentiert.

Tabelle 6.1: Bewertung des beispielhaften Modellierungsansatzes

Tabelle 6.2 zeigt die Ergebnisse der Bewertung im Vergleich mit den in Kapitel 4 bewerteten Ansätzen. Daraus ist zu erkennen, dass der als Anwendungsbeispiel entwickelte Ansatz alle Anforderungen zumindest teilweise erfüllt und insgesamt besser für Umgestaltungsprojekte geeignet ist als alle anderen bewerteten Ansätze. Allerdings werden Vier Anforderungen nur teilweise vom Anwendungsbeispiel erfüllt:

- Um die Anforderungen 8 (Brauchbarkeit) und 13 (Status der Modelle) voll zu erfüllen, hätten existierende Modelltypen und Modellelementtypen angepasst werden müssen. Davon hat der Autor abgesehen, weil die Erweiterung mit dem unveränderten ADOit kompatibel sein sollte. Daher erfüllen nur die neu zum Ansatz hinzugefügten Modellelemente und der Modelltyp „Kooperationsszenario“ die beiden Anforderungen voll.
- Um Anforderung 11 zu erfüllen, wurden nach dem Beispiel von ArchiMate mehrere Einsatzszenarien umrissen. Um die Anforderung als voll erfüllt werten zu können, müsste der beispielhafte Modellierungsansatz eingesetzt und die Beschreibung der Einsatzszenarien mit den gewonnenen Erfahrungswerten ergänzt werden.
- Ebenso hätte Anforderung 12 (Modellmanipulation) nur durch Rückkopplung mit der Praxis voll erfüllt werden können.

6.2.3 Praxistauglichkeit

Der erste Indikator für die Praxistauglichkeit ist die Integration mit einem unternehmenstauglichen Modellierungswerkezeug. Dies wurde bereits durch das Anwendungsbeispiel gezeigt (siehe Abschnitt 5.3). Darin werden die Bausteine auf einen

Anforderung	Archi-Mate	ARIS	BEN	EAM-Patterns	eGPM	Beispiel
1 IT-gestützte Arbeitsgeg.	●	–	–	●	+	+
2 Manuelle Tätigkeiten	●	●	●	–	+	+
3 Koordination	●	–	–	●	+	+
4 Fachliche Konzepte	+	+	+	●	+	+
5 Fachliche Architektur	+	–	+	+	+	+
6 Technologische Sichten	+	–	–	+	–	+
7 Beziehungen zw. Sichten	+	●	+	●	+	+
8 Zweifel an Brauchbarkeit	–	–	–	–	+	●
9 Validierung	–	–	–	–	+	+
10 Vages Modellieren	–	●	–	–	+	+
11 Einsatzszenarien	+	●	+	+	●	●
12 Modellmanipulation	–	●	–	+	●	●
13 Status der Modelle	–	●	–	–	+	●

Tabelle 6.2: Vergleich der bewerteten Modellierungsansätze

Ansatz mit einem Software-gestützten, für den Unternehmenseinsatz geeigneten Modellierungswerkzeug angewendet – ADOit.

Für den zweiten Indikator – Modellierung eines Umgestaltungsszenarios – wird auf ein reales Projekt zurückgegriffen. Einer der Projektmitarbeiter verfasste im Projektkontext eine Diplomarbeit (siehe [Ros09]), in der das Projekt so ausführlich beschrieben ist, dass dafür ex-post Modelle mit dem beispielhaften Modellierungsansatz erstellt werden konnten. Der Projektmitarbeiter erklärte sich bereit, die vom Autor erstellten Modelle anhand eines Fragebogens zu bewerten.

Das Projekt wurde schon in Abschnitt 3.5 grob beschrieben. Die Pfefferminzia-Versicherung¹ wollte ein Workflow-Management-System (im Folgenden kurz *WfMS*) in ihre Anwendungslandschaft integrieren. Dazu musste das WfMS an mehrere Softwaresysteme der Anwendungslandschaft angebunden werden, was die Kopplung der Softwaresysteme veränderte. Das WfMS sollte die Bearbeitungszeit von Vorgängen wie z.B. der Schadensregulierung verkürzen und zudem die Auskunftsfähigkeit über

¹Da die Versicherung namentlich nicht genannt werden möchte, wird sie dieser Arbeit als *Pfefferminzia* bezeichnet.

den Status der Vorgänge erhöhen. Daher veränderte die Einführung des WfMS die Arbeitsabläufe der betroffenen Akteure, in dem es beispielsweise bislang manuelle Arbeitsschritte automatisierte. Das Vorhaben der Versicherung kann demnach als „Umgestaltung“ nach Definition 2.3.1 bezeichnet werden.

Die Diplomarbeit beschreibt einen frühen Teil des Projekts, nämlich zwei „Proof-of-Concept“ mit verschiedenen WfMS. Die Anbieter der WfMS sollten anhand ausgewählter Geschäftsprozesse zeigen, wie gut sich ihre WfMS für die Versicherung eignen. Die dazu notwendigen Tätigkeiten stellen typische Umgestaltungstätigkeiten (siehe Abschnitt 2.3) dar: Die Projektmitarbeiter mussten unter anderem technische und fachliche Aspekte der Anwendungslandschaft erfassen, Geschäftsprozesse neu gestalten und die WfMS technisch integrieren. Daher kann das Projekt der Pfefferminzia als Umgestaltungsprojekt bezeichnet werden.

In der Diplomarbeit des Projektmitarbeiters wurden die beiden Proof-of-Concept anhand eines Kriterienkataloges miteinander verglichen. Das Ergebnis des Vergleichs floss in die Entscheidung der Versicherung ein, mit welchem der beiden Anbieter das Projekt umgesetzt werden sollte.

Um eine fundierte Entscheidung zu treffen, musste die Versicherung relevante, nicht-triviale Geschäftsprozesse für die beiden Proof-of-Concept auswählen. Dazu mussten Fragen wie folgende beantwortet werden:

- Welche Geschäftsprozesse profitieren besonders von einem WfMS?
- Wie würden sich die Arbeitsabläufe in diesen Geschäftsprozessen durch ein WfMS verändern?
- Gibt es in den Geschäftsprozessen wiederkehrende Teilprozesse, die durch ein WfMS automatisiert werden können? Solche Teilprozesse sind für die Versicherung besonders interessant, weil ihre Implementierung im WfMS für mehrere Geschäftsprozesse wiederverwendet werden kann.

Die Versicherung beantwortete diese Fragen mit Hilfe des Modellierungsansatzes eGPM (siehe 4.2.5). Mehrere Soll-Modelle beschreiben den gewünschten Einsatz eines WfMS. Diese Modelle sind in [Ros09] abgebildet und wurden vom Autor mit dem erweiterten ADOit nachmodelliert (Abbildungen siehe Anhang E):

- Antragsprozess für eine Krankenversicherung: [Ros09, S.54] und E.1
- Regelprüfung im Antragsprozess: [Ros09, S.55] und E.2

- Prozess für eine Ehegattenermäßigung: [Ros09, S.56] und E.3
- Krankenleistungsprozess: [Ros09, S.58] und E.4

Die fachlichen Konzepte, die in diesen Geschäftsprozessen von Bedeutung sind, wurden in einem Modell des Typs Datenarchitektur abgebildet (E.5).

Neben den Geschäftsprozessen werden in [Ros09, S.10ff] die Anwendungslandschaft des Versicherers, die eingesetzten Technologien und einige Services beschrieben, die von einem WfMS angeboten werden müssen. Ein Modell zeigt eine Bausteinsicht auf die zukünftige Anwendungslandschaft und dient als Referenz dafür, wie bestehende Softwaresysteme und ein WfMS integriert werden sollen. Aus diesen Angaben hat der Autor Modelle der Typen Anwendungsarchitektur (E.6) und Technologiearchitektur (E.7) erstellt.

Alle vom Autor erstellten Modelle wurden dem Projektmitarbeiter vorgelegt und vom Autor erklärt. Anhand des als Tabelle 6.3 abgebildeten Fragebogens bewertete der Projektmitarbeiter die Praxistauglichkeit des vorgestellten Modellierungsansatzes. Insgesamt erschien ihm der vorgestellte Ansatz eher besser geeignet als der im Projekt verwendete.

Damit deuten beide Indikatoren des Qualitätsmerkmals „Praxistauglichkeit“ – die Integration in ein unternehmenstaugliches Modellierungswerkzeug und die Modellierung eines Umgestaltungsszenarios – darauf hin, dass ein mit den Bausteinen angepasster Modellierungsansatz für den Unternehmenseinsatz geeignet ist.

6.2.4 Fazit

Zusammengefasst erfüllen die Bausteine das Ziel, eine Verbesserung der Modellierung in Umgestaltungsprojekten zu erreichen. Die Bausteine sind. . .

... **relevant**, weil sie die Modellierungsprobleme minimieren. Sie berücksichtigen die grundlegenden Eigenschaften der Modellierung in Umgestaltungsprojekten und helfen, die Anforderungen zu erfüllen, die in solchen Projekten an Modellierungsansätze zu stellen sind.

... **anwendbar**, weil mit ihnen ein Modellierungsansatz angepasst wurde.

... **praxistauglich**, weil der angepasste Modellierungsansatz auf einem verbreiteten, kommerziellen und werkzeuggestützten Ansatz aufbaut. Mit dem erweiterten Ansatz konnte ein realistisches Umgestaltungsszenario modelliert werden.

	schlechter	eher schlechter	gleich gut	eher besser	besser
Vergleichen Sie das erweiterte ADOit mit dem im Projekt eingesetzten Ansatz:					
Wie gut lässt sich die Referenzarchitektur mit dem Ansatz darstellen?					X
Wie gut hätte sich der Ansatz für das Umgestaltungsprojekt insgesamt geeignet?				X	
Wie gut hätten sich folgende Fragen mit dem erweiterten ADOit beantworten lassen?					
Welche Geschäftsprozesse profitieren besonders von einem WfMS?				X	
Wie würden sich die Arbeitsabläufe in diesen Geschäftsprozessen durch ein WfMS verändern?				X	
Gibt es in den Geschäftsprozessen wiederkehrende Teilprozesse, die durch ein WfMS automatisiert werden können?			X		

Tabelle 6.3: Fragebogen zur Evaluation der Praxistauglichkeit

6.3 Zusammenfassung

In diesem Kapitel wurde bewertet, ob die Bausteine helfen, die Modellierung in Umgestaltungsprojekten zu verbessern. Dazu wurde ein Vorgehen eingesetzt, das auch in der Softwaretechnik und in der Modellierung angewendet wird, um komplexe Lösungen hinsichtlich ihrer Qualität zu bewerten: Das übergeordnete Ziel wurde in mehrere Qualitätsmerkmale aufgeschlüsselt. Für diese wurden Indikatoren festgelegt, anhand derer bewertet werden kann, ob die Lösung die Qualitätsmerkmale aufweist.

Damit die Bausteine ihren Zweck erfüllen können, müssen sie die drei Qualitätsmerkmale *Relevanz*, *Anwendbarkeit* und *Praxistauglichkeit* aufweisen. Als Indikatoren wurden mehrere Ergebnisse dieser Arbeit herangezogen: Die empirisch festgestellten Modellierungsprobleme, der Anforderungskatalog und der beispielhafte Modellierungsansatz. Zusätzlich wurde mit diesem Ansatz ein reales Beispiel modelliert. Damit wurde gezeigt, dass mit den Bausteinen relevante, praxistaugliche Modellierungsansätze entwickelt werden können.

Kapitel 7

Zusammenfassung und Ausblick

Unternehmen müssen ihre Anwendungslandschaften laufend aus geschäftlichen, technischen oder rechtlichen Gründen anpassen. Oft betreffen solche Veränderungen mehrere Softwaresysteme einer Anwendungslandschaft und wirken sich auf die Arbeitsabläufe der Anwender aus. Die Zusammenhänge zwischen den Softwaresystemen und den Arbeitsabläufen sind komplex. Projektleiter, Architekten, Softwareentwickler, Tester und weitere Personengruppen müssen den Ausgangs- und den Zielzustand der Anwendungslandschaft und der von ihr unterstützten Geschäftsprozesse verstehen. Nur so können die Beteiligten ihre Anwendungslandschaft erfolgreich umgestalten.

In der Praxis werden Modelle als Hilfsmittel für die Umgestaltung von Anwendungslandschaften eingesetzt. Sie sollen transparent machen, wie die Softwaresysteme technisch und fachlich zusammenhängen und welche Konsequenzen die geplanten Veränderungen haben. Dies trifft besonders auf Ansätze zur Unternehmensmodellierung zu, weil sie entwickelt wurden, um die Zusammenhänge zwischen Technik, Organisation, Geschäftsprozessen und strategischen Zielen in Modelle abzubilden. Die in dieser Arbeit analysierten Modellierungsansätze eignen sich jedoch nur eingeschränkt, um die Umgestaltung einer Anwendungslandschaft zu modellieren. Die festgestellten Schwächen umfassen unter anderem:

- Einige Ansätze können nicht darstellen, wie die Akteure eines Geschäftsprozesses kooperieren und wie ihre Tätigkeiten durch Softwaresysteme unterstützt werden.
- Zum Teil vermitteln die Ansätze den Eindruck, dass ein praktisch nicht erreichbarer Idealzustand angestrebt wird: Eine Anwendungslandschaft soll

vollständig und korrekt modelliert werden.

- Der Modellierungsprozess wird oft nur als Mittel zum Zweck verstanden – nämlich, fertige Modelle zu produzieren. Dadurch entgeht die Chance, die von der Umgestaltung betroffenen Menschen durch gemeinsames Modellieren miteinander kommunizieren und lernen zu lassen.

Das Ziel dieser Arbeit ist, die Anpassung existierender Ansätze zu ermöglichen, damit sie die Umgestaltung von Anwendungslandschaften besonders gut unterstützen. Dazu musste zunächst die Forschungsfragen beantwortet werden, was Umgestaltung charakterisiert und was in Umgestaltungsprojekten mit Modellen bezweckt wird. Diese beiden Fragen konnten durch eine empirische Untersuchung von Praxisprojekten beantwortet werden. Aus den Ergebnissen und anhand von Literatur wurden Definitionen für die Begriffe „Anwendungslandschaft“ und „Umgestaltungsprojekt“ entwickelt. Die Definitionen ermöglichen, über Modellierungsprobleme in Umgestaltungsprojekten zu diskutieren. Die untersuchten Projekte zeigten mehrere Probleme, die mittels eines Fragebogens von Experten bewertet wurden. Sechs relevante Modellierungsprobleme werden in dieser Arbeit geschildert, unter anderem:

- Modelle zeigen zwar, dass ein Softwaresystem einen Prozess unterstützt – aber nicht, wie das Softwaresystem einen Prozess unterstützt.
- Abhängigkeiten zwischen Softwaresystemen sind schwer abzubilden, wenn sie sich nicht aus technischen Schnittstellen ergeben.

An die Frage, welche Modellierungsprobleme es gibt, schließt sich unmittelbar eine weitere Frage an: Welche Gründe gibt es für die festgestellten Modellierungsprobleme? In einer Analyse wurden vier Eigenschaften ermittelt, welche die Modellierung in Umgestaltungsprojekten grundsätzlich prägen:

- Anwendungslandschaften und ihr Einsatz in Geschäftsprozessen können nur vage in Modelle abgebildet werden. Solche Modelle stellen keine Fakten dar, sondern eine „konstruierte Realität“. Die Brauchbarkeit der Modelle kann nicht objektiv, sondern nur von den Modellnutzern in Bezug auf den Modellzweck bestimmt werden.
- Modelle sind nicht bloß Hilfsmittel für die Umgestaltung, sondern genauso Gestaltungsgegenstand von Umgestaltungsprojekten wie die Anwendungslandschaft selbst. Modelle müssen in einem zyklischen Prozess entworfen,

eingesetzt und verbessert werden. Der Prozess wiederholt sich, bis die Modelle für ihren Zweck – etwa den Einsatz als „Bauplan“ – ausreichend sind.

- Modellierung muss als Lern- und Kommunikationsprozess verstanden werden, in dem die mit der Anwendungslandschaft arbeitenden Personen gemeinschaftlich Modelle gestalten. Dadurch verändert die Modellierung die modellierte Organisation.
- Anwendungslandschaften müssen von verschiedenen Standpunkten in Modelle abgebildet werden: Technologische Sichten zeigen die Softwaresysteme und ihre Abhängigkeiten. Fachliche Sichten zeigen, wie die Softwaresysteme in Arbeitsabläufen verwendet werden. Die verschiedenen Sichten auf eine Anwendungslandschaft werden durch Modelle ausgedrückt und sind durch das gemeinsame Ziel des Umgestaltungsprojekts miteinander verbunden.

Berücksichtigt ein Modellierungsansatz diese Eigenschaften, können Modellierungsprobleme in Umgestaltungsprojekten vermieden werden.

Um zu bewerten, wie gut sich existierende Modellierungsansätze für die Umgestaltung eignen, wurden aus den grundlegenden Eigenschaften 13 Anforderungen abgeleitet – Anforderungen an die Modellierungssprache, das Modellierungsvorgehen und an den Einsatz von Modellen beschreibt, beispielsweise:

- Manuelle und IT-gestützten Tätigkeiten müssen unterscheidbar sein.
- Der Modellierer muss ausdrücken können, an welchen Stellen eines Modells er an dessen Brauchbarkeit zweifelt.
- Modelle müssen bereits im Modellierungsprozess validiert werden.

Einer der bewerteten Ansätze ist die am Fachbereich Informatik der Universität Hamburg entwickelte exemplarische Geschäftsprozessmodellierung (eGPM). Um weitere Ansätze für die Bewertung zu finden, wurden Literaturstudien herangezogen und folgende Ansätze ausgewählt:

- Die Architektur integrierter Informationssysteme (ARIS).
- Unternehmensmodellierung nach dem Business Engineering-Ansatz (BEN).
- Die Enterprise Architecture Management Patterns der TU München.

- ArchiMate (eine von der Open Group spezifizierte Modellierungssprache).

Zusammengefasst eignen sich eGPM, ArchiMate und die EAM-Patterns am besten für Umgestaltungsprojekte. Aber selbst diese Ansätze erfüllen nur einen Teil der Anforderungen. In der Praxis gibt es daher Bedarf an Modellierungsansätzen, die sich besser für Umgestaltungsprojekte eignen. Da aber jede Anforderung durch mindestens einen der Ansätze erfüllt wurde, gibt es Sprachmittel und Vorgehensweisen, die zur Konstruktion besser geeigneter Ansätze wiederverwendet werden können.

Der Anforderungskatalog war ein erster Schritt, um die letzte Forschungsfrage zu beantworten: Welche Mittel muss ein Modellierungsansatz mitbringen, damit er sich besonders für Umgestaltungsprojekte eignet? Aus den Ergebnissen der Bewertung wurden geeignete Sprachmittel und Vorgehensweisen identifiziert und verallgemeinert, sodass sie auch außerhalb ihres ursprünglichen Ansatzes angewendet werden können. So zum Beispiel das Konzept, Geschäftsprozesse Szenario-basiert zu modellieren. In dieser Arbeit werden zehn solcher Konzepte als „Bausteine“ beschrieben. Ein Anwendungsbeispiel zeigt, wie mit den Bausteinen ein Modellierungsansatz angepasst werden kann. In diesem Beispiel wurde der kommerzielle, werkzeuggestützte Ansatz *ADOit* mit Hilfe mehrerer Bausteine angepasst. Somit beantwortet diese Arbeit deskriptiv und konstruktiv die Frage, welche Mittel ein Modellierungsansatz mitbringen muss, damit er sich besonders für Umgestaltungsprojekte eignet.

Umgestaltungsprojekte wurden bisher in der Praxis und Forschung nicht als eigenständiger Typ von Projekt wahrgenommen, sondern z.B. als Software-Entwicklungsprojekte oder als Migrationsprojekte betrachtet. Der in dieser Arbeit eingeführte Projekttyp *Umgestaltungsprojekt* leistet daher einen Beitrag zur Weiterentwicklung der (Wirtschafts-)Informatik. Dieser Beitrag ist besonders durch seine anwendungsorientierte Sichtweise gekennzeichnet, die in der informatischen Modellierung wenig berücksichtigt wird. Im Zentrum dieser Sichtweise steht die Frage, wie Softwaresysteme die Menschen bei ihren betrieblichen Aufgaben unterstützen und damit IT-gestützte Geschäftsprozesse ermöglichen. Den Anwendern der Softwaresysteme wird bei dieser Betrachtungsweise besondere Aufmerksamkeit zu Teil. Das schlägt sich in den Bausteinen nieder, welche die Modellierungssprache und die Vorgaben zum Modellierungsvorgehen umreißen. Die Anwender sollen in die Lage versetzt werden, mit anderen Akteuren gleichberechtigt über Modelle zu kommunizieren.

Ein weiterer innovativer Aspekt dieser Arbeit ist, dass unvollständige, falsche Modelle nicht abgelehnt, sondern als Gegebenheit akzeptiert werden. Es ist Auf-

gabe eines Modellierungsansatzes, diese Gegebenheit zu berücksichtigen und die Modellierer dabei zu unterstützen, trotzdem brauchbare Modelle zu erstellen.

Die Evaluation der Bausteine zeigte, dass sie relevant, anwendbar und praxistauglich sind und daher die vom Autor festgelegten Qualitätsmerkmale aufweisen. Damit erreicht diese Arbeit ihr Ziel: Durch besser für Umgestaltungsprojekte geeignete Modellierungsansätze eine Verbesserung der Praxis zu ermöglichen. Trotzdem gibt es noch Potential zur Verbesserung, das durch weitere Arbeiten gehoben werden kann:

- Diese Arbeit beschreibt zehn Bausteine. Quelle der Bausteine war die Bewertung bestehender Ansätze. Es gibt zwei Wege, um zusätzliche Bausteine zu finden: Der Anforderungskatalog kann um weitere Anforderungen ergänzt werden und weitere Modellierungsansätze können in die Bewertung aufgenommen werden.
- Bisher gibt es nur ein einziges Anwendungsbeispiel für die Bausteine. Um anderen Forschern und Praktikern den Zugang zum Thema zu erleichtern und die Diskussion über Umgestaltungsprojekte voranzutreiben, können weitere Ansätze mit den Bausteinen adaptiert werden.
- Der praktische Einsatz der Bausteine und eines damit entwickelten Ansatzes steht noch aus.
- Die Bausteine können um eine Methode ergänzt werden, mit der Unternehmen maßgeschneiderte Modellierungsansätze für ihre Umgestaltungsprojekte entwickeln können. Ein Beispiel für eine vergleichbare Methode liefert *TOGAF* (*The Open Group Architecture Framework*, siehe [The09]), das im Bereich der Unternehmensmodellierung verbreitet ist. *TOGAF* beschreibt eine *Architecture Development Method*, die dabei anleitet, einen Ansatz zur Modellierung einer Unternehmensarchitektur zu erstellen.

Neben der Erweiterung und Anwendung der Bausteine gibt es in dieser Arbeit noch andere Anknüpfungspunkte für die wissenschaftliche Auseinandersetzung mit Umgestaltungsprojekten:

- Ein für Umgestaltungsprojekte geeignetes Vorgehensmodell muss ermöglichen, die Anwendungslandschaft und ihre Modelle wiederholt rückzukoppeln.

Umgestaltungsprojekte erfordern daher ein zyklisches Vorgehensmodell. Es erscheint dem Autor lohnenswert, ein zyklisches Vorgehensmodell für Umgestaltungsprojekte zu entwickeln. Ein solches könnte wesentlich zum strukturierten Ablauf von Umgestaltungsprojekt beitragen, so wie dies in vergleichbarer Weise in Software-Entwicklungsprojekten mit Vorgehensmodellen (z.B. *Scrum* [SS14]) erreicht wurde.

- Anwendungslandschaften werden in dieser Arbeit aus betriebswirtschaftlicher, technologischer und aus anwendungsorientierter Perspektive betrachtet (vgl. Abschnitt 1.2). Besonders aus anwendungsorientierter Perspektive zeigen sich die organisatorischen Auswirkungen der Umgestaltung von Anwendungslandschaften. Die Wechselwirkung zwischen technischen Änderungen, Geschäftsprozessen und der Struktur einer Organisation wurde in Abschnitt 2.3 als *Co-Evolution* bezeichnet. Das Zusammenwirken von Individuen, Organisation und Technik stellt nicht nur für die Informatik, sondern auch für die Organisationstheorie ein interessantes Themengebiet dar. Daher scheint es vielversprechend, die Umgestaltung von Anwendungslandschaften auch interdisziplinär zu untersuchen.
- Modelle als Lern- und Kommunikationsmittel einzusetzen, birgt das Risiko, einige wenige *Modell-mächtige* Akteure weiter zu stärken (vgl. [Bra73] und Abschnitt 3.4). Um ein Modellmonopol der Modellierer zu vermeiden und einen „demokratischen“ Zugang zu Modellen und ihrem Nutzen zu ermöglichen, darf die Modellierungssprache eines Ansatzes nicht nur für Experten anwendbar sein. Dieses Problem wird im Rahmen der Qualität von Modellen wissenschaftlich diskutiert (vgl. beispielsweise [MDN08] und [MRC07]). Modellierungsansätze für Umgestaltungsprojekte können von Erkenntnissen dieses Forschungsgebiets profitieren.

Diese Dissertation wurde durch Probleme motiviert, die der Autor in seiner eigenen Praxis als IT-Berater erlebt hat. Der Praxisbezug zieht sich durch dieser Arbeit. Unter anderem wurden. . .

- Umgestaltungsprojekte empirisch untersucht,
- festgestellte Modellierungsproblemen durch Experten beurteilt,
- praxisrelevante Modellierungsansätze bewertet und

- ein Anwendungsbeispiel auf Basis eines kommerziellen Modellierungsansatzes entwickelt.

Darin begründet sich die Hoffnung des Autors, dass mit dieser Arbeit nicht nur ein wissenschaftlicher Beitrag geleistet wurde, sondern auch Praktiker daraus Nutzen ziehen können.

Anhang A

Empirische Untersuchung von Praxisprojekten

Die empirische Untersuchung von Praxisprojekten war Ausgangspunkt der Begriffsbildung. Daher weichen die in diesem Anhang verwendeten Begriffe zum Teil von ihrer im Hauptteil definierten Bedeutung ab.

A.1 Experteninterviews

Als ersten Schritt zur Begriffsbildung erarbeitete der Autor aus seiner eigenen Projekterfahrung und aus veröffentlichten Projektbeschreibungen anderer Autoren mehrere sogenannter „Umgestaltungsszenarien“. Diese beschreiben Tätigkeiten, Rollen und Probleme in Projekten, in denen Anwendungslandschaften verändert wurden. Zur Veranschaulichung wurden Kooperationsbilder des Modellierungsansatzes eGPM eingesetzt (siehe Anhang C). Die folgenden Abschnitte wurden fünf Experten in schriftlicher Form vorgelegt. Die Experten waren als Softwareentwickler, Softwarearchitekt, Projektleiter und Berater an den untersuchten oder vergleichbaren Projekten beteiligt waren.

Die Auswertung erfolgte durch halbstrukturierte Interviews. In den Interviews wurden die Experten darum gebeten, aus ihrer Projekterfahrung heraus weitere Umgestaltungsszenarien vorzuschlagen. Dadurch stieg die Anzahl an untersuchten Umgestaltungsszenarien von ursprünglich fünf auf sieben.

Tabelle A.1 zeigt die Ergebnisse der Auswertung.

Umgestaltungsszenario	relevant	tw. relevant	nicht relevant	k. A.
Testfälle	5	0	0	0
Handlungsbedarf	5	0	0	0
Sichten	4	1	0	0
Fahrplan	4	1	0	0
Planungsperioden Sichten	4	0	0	1
Migrationsarchitektur	2	1	0	2
Bewertung	2	0	0	3

Tabelle A.1: Bewertung der Umgestaltungsszenarien durch Experteninterviews

A.1.1 Einleitung

Die folgenden Umgestaltungsszenarien beschreiben Tätigkeiten, welche die Umgestaltung von Anwendungslandschaften charakterisieren und damit den Begriff der „Umgestaltung“ von Anwendungslandschaften verdeutlichen. Die Umgestaltungsszenarien bestehen aus den beiden Abschnitten „Kontext“ und „Szenario“, sowie je einer Abbildung, die die wesentlichen Akteure und Gegenstände der Szenarien darstellt. In den Umgestaltungsszenarien wird keine Bewertung der geschilderten Tätigkeiten vorgenommen. Lösungen für mögliche Probleme und Maßnahmen zur Unterstützung der Akteure sind ebenfalls *nicht* Bestandteil der Umgestaltungsszenarien.

Die Umgestaltungsszenarien fundieren auf mehreren Quellen: Auf meiner eigenen Projekterfahrung, auf den Erfahrungen von Interviewpartnern und auf Fachliteratur. Tabelle A.2 beschreibt, welchen Quellen in den einzelnen Umgestaltungsszenarien herangezogen wurden. Die als Quellen verwendeten Projekte wurden von Finanzdienstleistern, einem Logistikunternehmen und einem Großhandel durchgeführt. Die in den Projekten geleisteten Aufwände reichen von wenigen tausend bis zu mehreren zehntausend Personentagen.

Folgende Definitionen sollen das Verständnis der Umgestaltungsszenarien erleichtern:

Anwendungslandschaft Die Gesamtheit der IT-Systeme einer Organisation sowie die Abhängigkeiten zwischen diesen Systemen.

Transformationsprojekt Ein Projekt, das tiefgreifende Änderungen in der Anwendungslandschaft und damit einhergehende Veränderungen in der Organisation

Szenario	Projekt	Rolle des Autors	Literatur
Testfälle	Projekt A	Testverantwortlicher	–
Handlungsbedarf	Projekt B	mit Detailanalyse beauftragter Mitarbeiter	–
Sichten	Projekt A	Koordinator der Anwendungsmigration	–
Fahrplan	Projekt A	Koordinator der Anwendungsmigration	[EHH ⁺ 08]
Planungsperioden	–	–	[MW04]
Migrationsarchitektur	Projekt C	–	[EHH ⁺ 08] [MW04]
Bewertung einer Anwendungslandschaft	Projekt D	Berater	–

Tabelle A.2: Quellen

eines Unternehmens umsetzt.

Migration Ein technischer Aspekt von Transformationsprojekten. Anwendungen oder Daten werden in eine andere technische Umgebung übergeführt.

A.1.2 Umgestaltungsszenario *Testfälle*

Kontext

Ein Finanzdienstleister tauscht an sämtlichen Arbeitsplätzen die bisher verwendeten PCs durch Thin Clients aus. Dieser Schritt beinhaltet auch einen Wechsel des Betriebssystems und das Outsourcing des IT-Hostings an einen Dienstleister. Da viele der Geschäftsprozesse unverändert bleiben sollen, müssen die sie unterstützenden Anwendungen auf die neue technische Basis migriert werden. Bei den Anwendungen handelt es sich um 30 zugekaufte Softwaresysteme (Standardprodukte und Individuallösungen). Die hauseigene IT des Finanzdienstleisters hat nur Betriebsaufgaben und bestimmt für jede Anwendung einen Anwendungsbetreuer, der die jeweilige Migration verantwortet.

Der Projektleiter setzt einen Testkoordinator ein, der durch Konzeption, Organisation und Controlling der Testaktivitäten die Migration der Anwendungen absichert.

Neben technischen Tests, welche die Lauffähigkeit der migrierten Anwendungen in der neuen Umgebung prüfen, erachtet der Testkoordinator auch fachliche Tests als notwendig.

Szenario

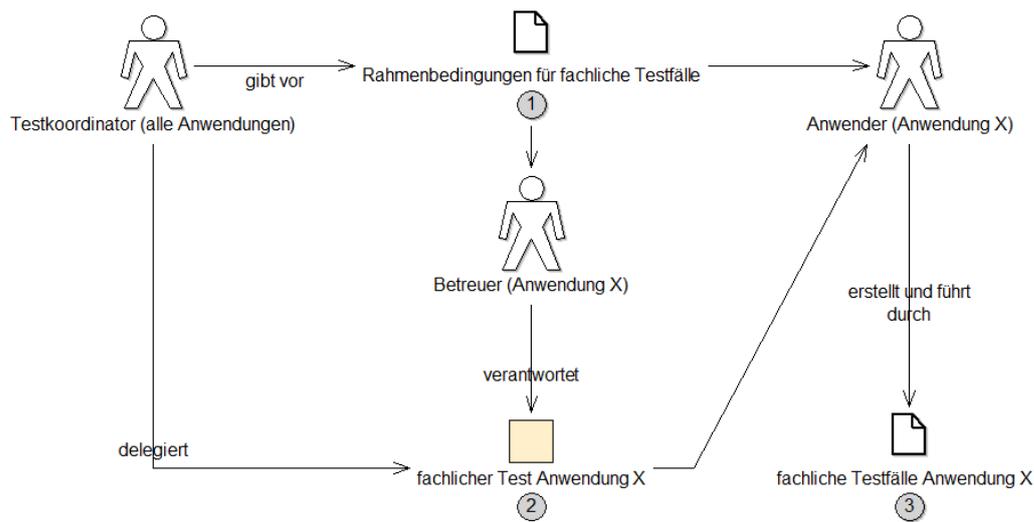


Abbildung A.1: Szenario „Testfälle“

Der Testkoordinator stellt den Anwendungsbetreuern die Rahmenbedingungen für die Testdurchführung vor. Da die Anwendungsbetreuer nicht über die nötigen Kenntnisse verfügen, ziehen sie für die fachlichen Tests Anwender hinzu. In kleinen Gesprächsrunden besprechen Testkoordinator, Anwendungsbetreuer und Tester je Anwendung durch, welche Veränderungen wann geplant sind und wie methodisch beim Testen vorgegangen werden soll. Die Tester erstellen daraufhin die fachlichen Testfälle.

Sobald der Projektplan den Einsatz einer neuen Programmversion oder die Änderung einer Schnittstelle vorsieht, fordert der Testkoordinator einen Nachweis über erfolgreiche Testdurchführung für die betroffene Anwendung ein. Nachdem die ersten Anwendungen getestet wurden, stellt der Testkoordinator fest, dass die Testfälle keine anwendungsübergreifenden Geschäftsprozesse berücksichtigen. Auf Nachfrage bei den Anwendungsbetreuern erfährt der Testkoordinator, dass weder sie noch die Tester einen Überblick darüber haben, welche Geschäftsprozesse sich über mehrere Anwendungen erstrecken.

A.1.3 Umgestaltungsszenario *Handlungsbedarf*

Kontext

Siehe Umgestaltungsszenario „Testfälle“.

Szenario

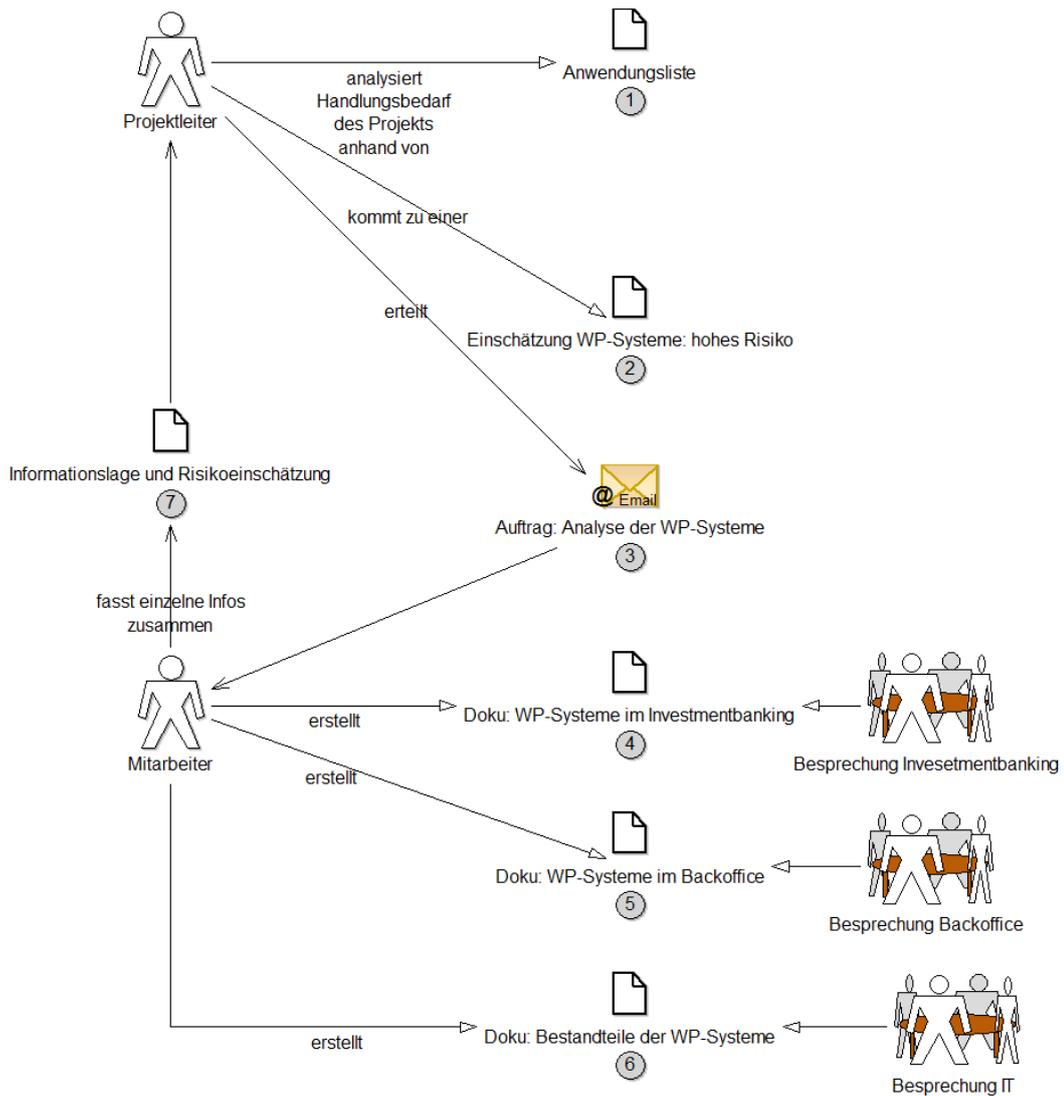


Abbildung A.2: Szenario „Handlungsbedarf“

Der Projektleiter bewertet mit seinem Team den Handlungsbedarf und die Risiken für die Migration der Anwendungen des Unternehmens. Für den Teil der Anwendungslandschaft, der das Wertpapiergeschäft unterstützt, kann kein befriedigendes

Ergebnis erzielt werden. Der Geschäftsbereich wurde als unternehmenskritisch identifiziert, sodass die Migration möglichst problemlos laufen muss. Aus der Anwendungsliste des Projekts, der gesichteten Dokumentation und den Rückfragen an den Fachbereich gingen nur rudimentäre Informationen hervor: Neben einer zugekauften Standardsoftware wird auch noch eine Eigenentwicklung eingesetzt, diese beiden Systeme arbeiten intensiv zusammen und das Wertpapiergeschäft wird von zwei Abteilungen abgewickelt. Einen konkreten Handlungsbedarf kann der Projektleiter daraus nicht ableiten. Das Risiko für den Bereich „Wertpapiersysteme“ wird daher als hoch eingestuft und der Projektleiter beauftragt einen seiner Mitarbeiter mit einer Analyse. Als Aufwand werden dafür fünf Personentage angesetzt.

Der Mitarbeiter findet durch seine Vernetzung im Haus jeweils zwei Personen aus den Abteilungen „Investmentbanking“ und „Backoffice“, die mit den zu untersuchenden Anwendungen arbeiten. In zwei separaten Besprechungen befragt er die Vertreter der Fachabteilungen nach den Arbeitsabläufen im Wertpapiergeschäft. Dadurch verschafft sich der Mitarbeiter einen groben Überblick darüber, welche Anwendungen von den Fachbereichen überhaupt wahrgenommen werden, welchen Zweck diese erfüllen und was sie miteinander zu tun haben. Das dokumentiert er in einem Gesprächsprotokoll. Es gibt keine bestehenden Geschäftsprozessmodelle. Der Mitarbeiter sieht auch davon ab, selbst solche Modelle mit den Vertretern der Fachabteilungen zu erstellen, da ihm für den Einstieg in das Thema die grobe Beschreibung reicht.

Als nächstes trifft sich der Mitarbeiter mit einem Kollegen aus der IT, der die Wertpapiersysteme kennt und sie zum Teil betreut. Bestehende Dokumentation gibt es fast nur zur zugekauften Anwendung – in Form von Unterlagen des Herstellers. In dem Gespräch wird klar, dass die in der Anwendungsliste als zwei Anwendungen vermerkten Systeme eigentlich aus acht Bestandteilen (Server-Anwendungen, mehrere Client-Programme, Middleware) bestehen. Wieder notiert der Mitarbeiter die Informationen in Form eines Gesprächsprotokolls.

Zum Abschluss der Analyse führt der Mitarbeiter die in den drei Terminen gesammelten Informationen ohne nochmalige Rücksprache mit Fachabteilungen oder IT zusammen. Dabei konsolidiert er die unterschiedlichen Wahrnehmungen der Systeme (IT-Sicht und zwei Sichten aus Fachabteilungen) in einer Überblicksgrafik und schätzt ab, welche Teile der Anwendungslandschaft im Projekt verändert werden. Die Informationsbasis weist aber noch Lücken auf, so dass der Zusammenhang

von Anwendungen und ihrem Zweck schwierig zu verstehen ist. Z. B. wird nicht klar, welche Rolle die Systeme für die Zusammenarbeit der beiden Fachabteilungen spielen. Außerdem hat der Mitarbeiter kein Vertrauen in den Überblick der Bestandteile der IT-Systeme. Dieser Informationsstand reicht noch nicht aus, um die für das Migrationsprojekt nötigen Entscheidungen zu treffen, beispielsweise welche Anpassungen bei Herstellern beauftragt werden müssen. Alle diese Informationen schreibt er in einen dreiseitigen Managementbericht für den Projektleiter, dem er eine weitere Detaillierung der Analyse empfiehlt.

A.1.4 Umgestaltungsszenario *Sichten*

Kontext

In einem Transformationsprojekt werden Anwendungen abgeschaltet, auf eine neue technische Basis migriert oder neu eingeführt. Grundlage für Bestimmung des Handlungsbedarfs war eine Anwendungsliste, die vom IT-Servicemanagement des Unternehmens erstellt wurde. Der Umbau der Anwendungslandschaft erfordert verschiedene technische Maßnahmen, wie beispielsweise die Anpassung von Schnittstellen, Datenmigrationen, Testphasen oder Veränderungen von Geschäftsprozessen. Ein Koordinator hat die Aufgabe, für eine anwendungsübergreifende Abstimmung der einzelnen Umbaumaßnahmen zu sorgen. Der in die Abstimmung einzubeziehende Personenkreis umfasst die Anwendungsbetreuer (verantworten die Umsetzung der Änderungen), Fachabteilungen (stellen Tester und Pilotanwender), die Personalabteilung (Vergabe von Berechtigungen), den zentralen IT-Dienstleister (Betrieb der Anwendungen, Bereitstellung einer Störungs-Hotline), die leitenden Personen im Projekt (Abstimmung mit parallel laufenden Tätigkeiten) und Hersteller von zugekaufter Software. Die technischen Umstellungsmaßnahmen wie beispielsweise Datenmigrationen führen die Anwendungsbetreuer selbst durch oder arbeiten dazu mit den Herstellern oder dem IT-Dienstleister zusammen.

Szenario

Der Koordinator verwendet die bestehende Anwendungsliste als Grundlage für seine Aufgabe. Anwendung für Anwendung spricht der Koordinator den jeweils verantwortlichen Anwendungsbetreuer und die übrigen Beteiligten an. Gemeinsam identifizieren sie z.B. den Zeitbedarf für Tests, die Notwendigkeit eines Pilotbetriebs

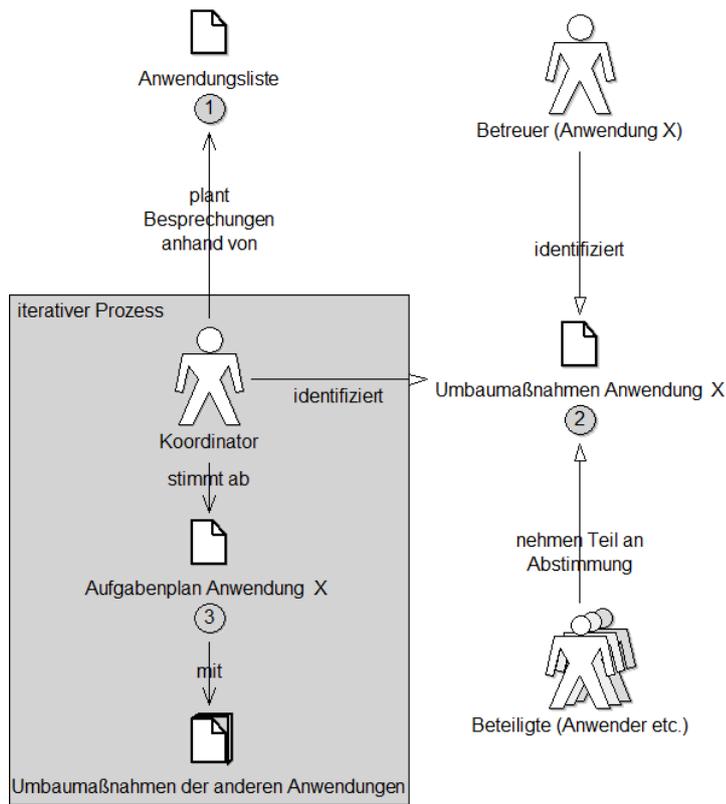


Abbildung A.3: Szenario „Sichten“

oder die Abhängigkeiten zu anderen Anwendungen. Dabei stellt der Koordinator fest, dass die Beteiligten unterschiedliche Auffassungen darüber haben, was als „Anwendung“ bezeichnet wird. Beispielsweise beobachtet der Koordinator folgende Sichtweisen auf die Office-Software:

- Ein Anwender bezeichnet jeden der drei Bestandteile der Office-Suite (Textverarbeitung, Tabellenkalkulation, Präsentationsprogramm) als eigene Anwendung.
- Ein Mitarbeiter des IT-Servicemanagements unterscheidet die Anwendungen anhand der vergebenen Lizenzen. Da zwei verschiedene Lizenztypen beschafft wurden (einmal mit Datenbankprogramm, einmal ohne), unterscheidet er zwei Anwendungen („Office-Suite Pro“, „Office-Suite Standard“).
- Ein Mitarbeiter des Dienstleisters, der mit der Installation der Software beauftragt ist, unterscheidet anhand der zu installierenden Programmpakete: Das Office-Paket selbst (unabhängig von der Lizenz handelt es sich um die gleiche Installationsdatei), ein Paket mit zusätzlichen Schriftarten und ein Paket mit Konvertern für verschiedene Dateiformate.

Um die Umbaumaßnahmen aller Anwendungen mit einander abstimmen zu können, muss der Koordinator diese verschiedenen Auffassungen erkennen und zwischen den Beteiligten vermitteln.

A.1.5 Umgestaltungsszenario *Fahrplan*

Kontext

Dieses Umgestaltungsszenario schließt unmittelbar an das Umgestaltungsszenario „Sichten“ (siehe A.1.4) an.

Die Anwendungslandschaft des Unternehmens ist technisch gesehen nur lose integriert. Zwar kommunizieren viele Systeme miteinander, doch verwenden sie dazu überwiegend Dateiaustausch mit zeitgesteuerter Verarbeitung (Batch). Deswegen ist es möglich, harte Schnitte bei der Migration zu setzen. Das heißt, dass im Regelfall kein Parallelbetrieb einer Anwendung in der alten und in der neuen technischen Umgebung nötig ist. Außerdem können alle Anwendungen stichtagsbezogen umgestellt werden. Es ist also nicht nötig, Anwendungen schrittweise über einen längeren Zeitraum zu migrieren.

Szenario

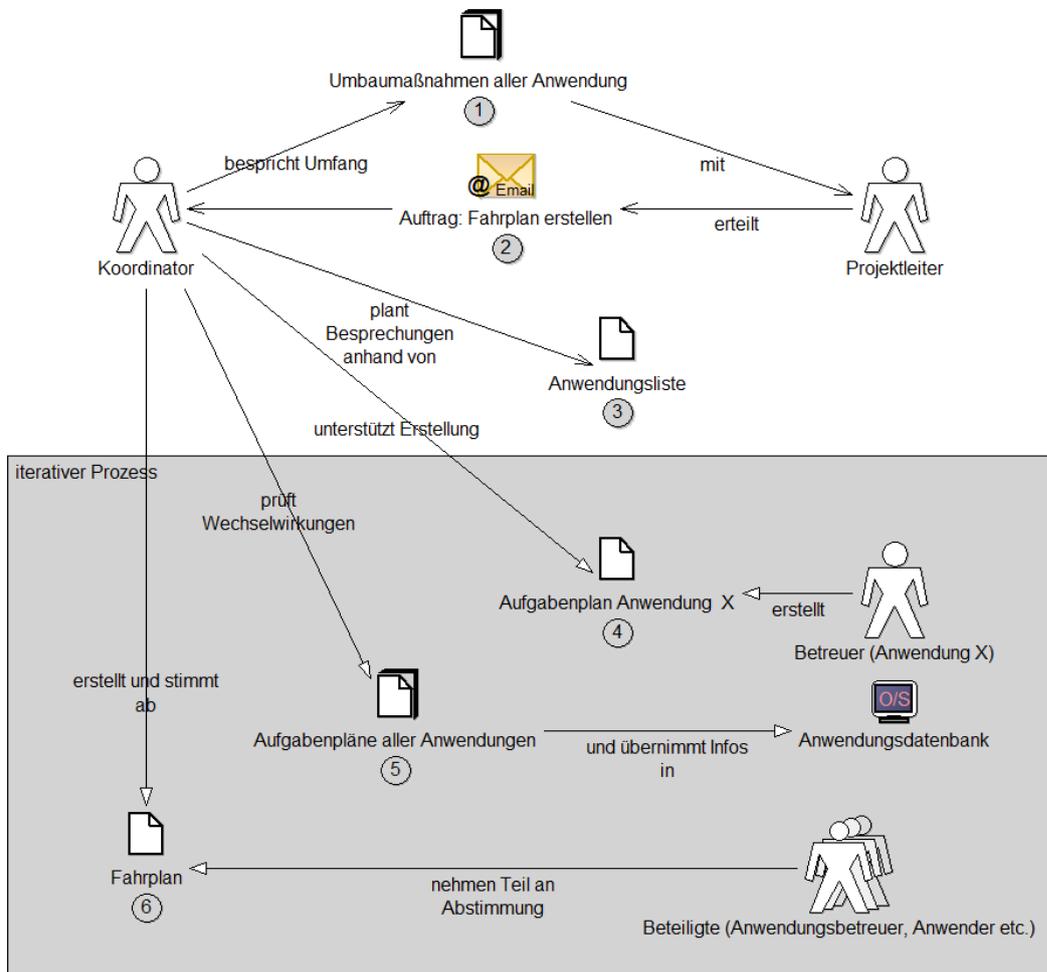


Abbildung A.4: Szenario „Fahrplan“

Auf Grund der großen Anzahl an Umbaumaßnahmen beschließen der Projektleiter und der Koordinator, die Inbetriebnahme der Änderungen auf vier Termine aufzuteilen. Der Projektleiter beauftragt den Koordinator, einen Fahrplan für die Inbetriebnahme zu erstellen.

Zunächst unterstützt der Koordinator die für die Änderungen verantwortlichen Personen dabei, die Maßnahmen je Anwendung in eine Reihenfolge zu bringen. Dann stimmt der Koordinator ab, welche Anwendungen zu welchem Termin in Betrieb gehen. Dazu muss er die technischen und fachlichen Abhängigkeiten zwischen den Anwendungen ermitteln und nochmals alle beteiligten Personen wegen der

Terminplanung mit einbeziehen. Um die anfallenden Informationen zu sammeln, baut der Koordinator die Anwendungsliste zu einer Datenbank aus.

Da einige der Gesprächspartner kaum Einblick in die geplanten Veränderungen der Anwendungslandschaft haben, erklärt der Koordinator in den Gesprächen einige der Zusammenhänge des Projekts. Außerdem vermittelt er bei Missverständnissen, die sich aus den unterschiedlichen Sichtweisen der Beteiligten ergeben (siehe Umgestaltungsszenario „Sichten“, Abschnitt A.1.4). Wenn sich dadurch neue Erkenntnisse über Abhängigkeiten zwischen Anwendungen ergeben, setzt der Koordinator weitere Abstimmungsrunden an.

A.1.6 Umgestaltungsszenario *Planungsperioden*

Kontext

Ein Unternehmen hat das strategische Ziel, die Zahl der Individual-Softwaresysteme zu reduzieren und diese durch Standard-Softwaresysteme zu ersetzen. Das Vorhaben erstreckt sich über fünf Jahre und damit über drei Planungsperioden der IT-Abteilung. In einer Planungsperiode legt das Unternehmen das IT-Budget und die personellen Ressourcen für zwei Jahre fest.

Szenario

Das Unternehmen initiiert ein Projekt, um das Vorhaben umzusetzen. Der Projektleiter erstellt mit seinem Team zunächst einen groben Gesamtprojektplan, den er mit der Planung des IT-Abteilungsleiters abstimmt. Gemeinsam legen sie fest, welche Teilprojekte in welchen Planungsperioden umgesetzt werden. Anschließend erstellen sie eine grafische Übersicht, welche Teile der Anwendungslandschaft in welchen Zeiträumen umgestaltet werden. Anhand dieses Modells diskutieren der Projektleiter und das Management des Unternehmens die Veränderungen, die durch das Vorhaben auf das Unternehmen zukommen.

A.1.7 Umgestaltungsszenario *Migrationsarchitektur*

Kontext

Nach dem Kauf neuer Maschinenanlagen muss ein Unternehmen deren Leitstandssoftware in die vorhandene Anwendungslandschaft integrieren. Die fachlich eng

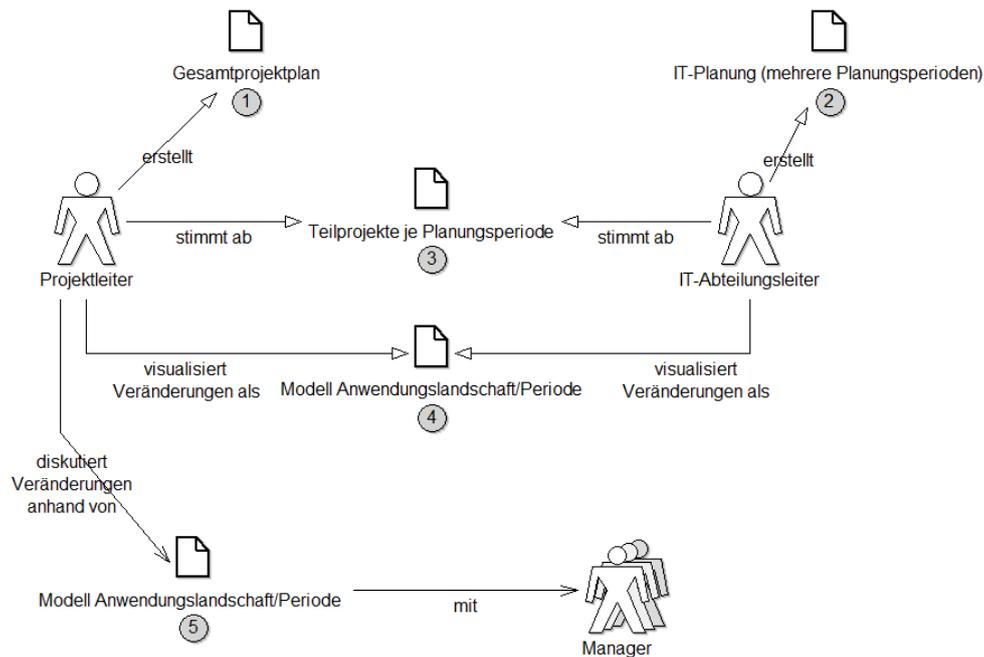


Abbildung A.5: Szenario „Planungsperioden“

verzahnte Anwendungslandschaft kommuniziert auf technischer Ebene Großteils per Messaging. Weil die Einführung auf Grund der zahlreichen technischen und fachlichen Abhängigkeiten ein komplexes und fehleranfälliges Unterfangen darstellt, bleiben die bisher verwendeten Anlagen zunächst in Betrieb. Ihre Aufgaben sollen schrittweise auf die neuen Anlagen übertragen werden, um die Risiken der Einführung zu minimieren. Ein Softwarearchitekt wird beauftragt, eine geeignete Migrationsarchitektur für die Inbetriebnahme auszuarbeiten, welche die technische und prozessuale Gestaltung der einzelnen Schritte beschreibt.

Szenario

Um ein Grundlage für die Migrationsarchitektur zu schaffen, trifft sich der Architekt mit den Anwendern der abzulösenden Leitstandssoftware. Gemeinsam modellieren sie die Prozesse, die durch die Anlagen und die Software unterstützt werden. Die dabei gesammelten Informationen über die benutzten Systeme ermöglichen es dem Architekten, ein erstes Modell des betroffenen Ausschnitts der Anwendungslandschaft zu erstellen. Dieses stimmt er mit den Anwendungsbetreuern aus der IT-Abteilung ab. Für die technischen Aspekte der Umgestaltung erachtet der

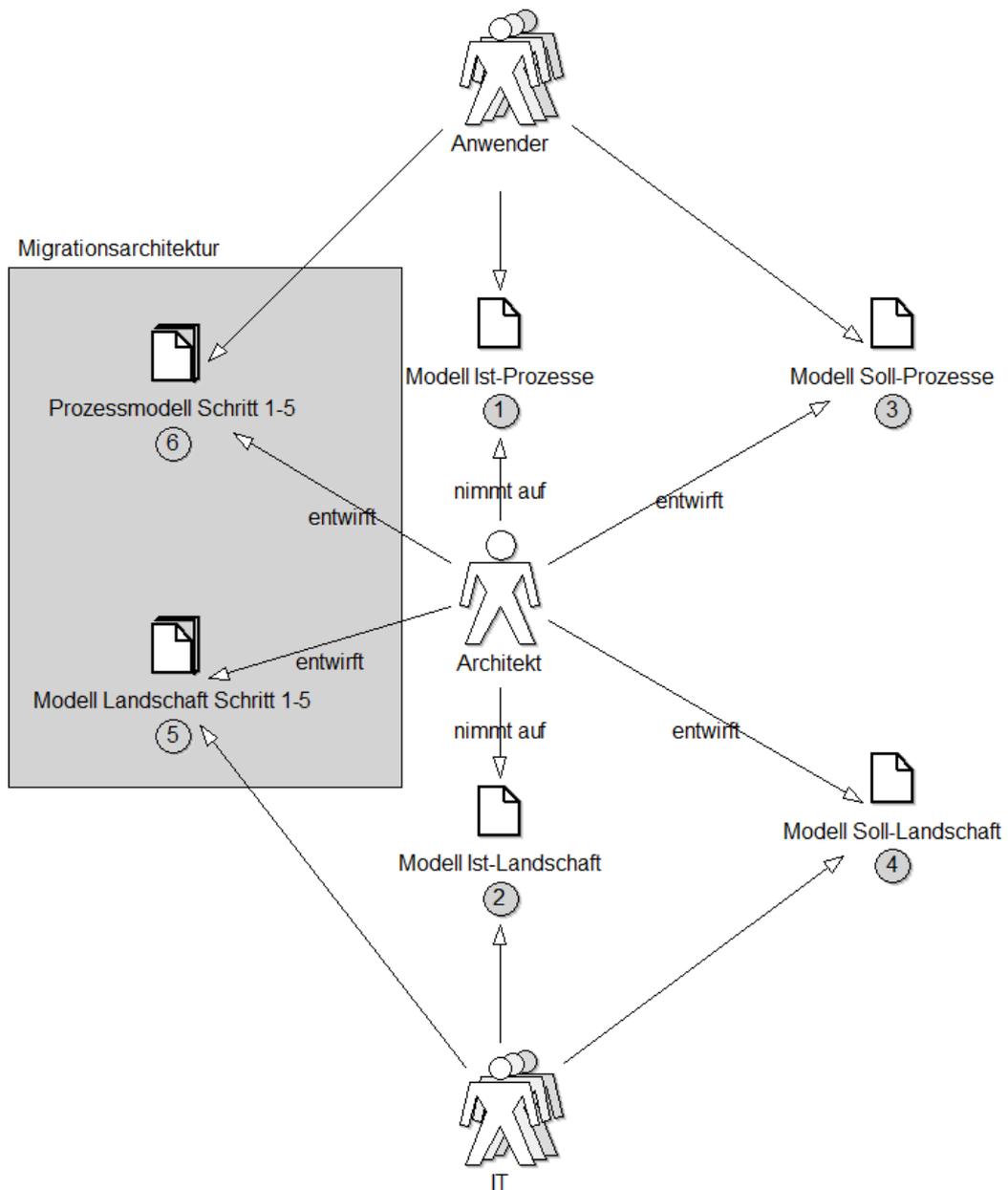


Abbildung A.6: Szenario „Migrationsarchitektur“

Architekt vor allem die einzelnen Kommunikationswege als auch die Art der Kommunikation zwischen den Systemen als relevant. Daher detailliert er gemeinsam mit den Anwendungsbetreuern das Modell um die Kommunikationsrichtungen, Art des Zugriffs und die Implementierungstechnologien der Schnittstellen.

Im nächsten Schritt modelliert der Architekt gemeinsam mit Anwendern und Anwendungsbetreuern den Sollzustand von Prozessen und Anwendungslandschaft. Nachdem ein Konsens über die Modelle für Ist- und Sollzustand erreicht wurde, entwirft der Architekt die Migrationsarchitektur. Dazu vergleicht er Ist- und Sollmodelle der Anwendungslandschaft, identifiziert die nötigen Umbaumaßnahmen und bringt sie in eine Reihenfolge. Der Architekt legt fünf Schritte zwischen Ist- und Sollzustand der Anwendungslandschaft fest, wobei mit jedem Schritt größere Anteile der Geschäftsprozesse von den alten auf die neuen Anlagen verlagert werden. Das bedeutet, dass die betroffenen Anwendungen in jedem Schritt angepasst werden müssen, indem beispielsweise Kommunikationswege über temporäre Adapter umgeleitet werden. Für jeden Schritt erstellt der Architekt ein Modell der Anwendungslandschaft, das er mit den Anwendungsbetreuern abstimmt. Nach der Ausarbeitung der technischen Aspekte der Migrationsarchitektur bespricht der Architekt jeden der Schritte mit den Anwendern und klärt, welche Auswirkungen die Umstellung auf die Verwendung der Systeme hat. Jedes Modell der Anwendungslandschaft wird mit einem korrespondierenden Modell der betroffenen Geschäftsprozesse ergänzt, das den veränderten Prozess dokumentiert. Nach mehreren Abstimmungsrunden mit Anwendern und Anwendungsbetreuern steht die Migrationsarchitektur mit ihren Schritten und den dazugehörigen technischen und fachlichen Modellen fest.

A.1.8 Umgestaltungsszenario *Bewertung einer Anwendungslandschaft*

Kontext

Die Führungskräfte eines Unternehmens haben wahrgenommen, dass in den Fachabteilungen große Unzufriedenheit mit der IT-Unterstützung herrscht. Daher wird ein Beratungsunternehmen beauftragt, die Anwendungslandschaft zu analysieren, Verbesserungspotential zu identifizieren und Lösungsvorschläge zu unterbreiten. Den Schwerpunkt der Analyse sollen das ERP-System (Standardsoftware mit Anpassungen) sowie ein Dutzend daran angeschlossene Systeme bilden.

Szenario

Da die Anwendungslandschaft schlecht dokumentiert ist, beginnt ein Team aus zwei Beratern des beauftragten Dienstleisters die Analyse mit einem Interview des IT-Leiters. Sie erfassen je Anwendung:

- Hersteller und Produktinformationen (z.B. Lizenzmodell),
- Typ (z.B. Webclient mit Server und Datenbank)
- eine kurze Historie (z.B. warum und wann die Anwendung beschafft wurde),
- wer die Anwendung benutzt,
- Kernaufgaben und bekannte Probleme,
- welche Daten gehalten werden,
- welche Schnittstellen das System aufweist und
- den Anwendungsbetreuer der IT.

Danach folgen Interviews mit einigen der Anwendungsbetreuer der IT-Abteilung, um Details zu den Anwendungen zu ergänzen. Das Ergebnis dokumentieren die Berater in einem Textdokument und mit einer Grafik, welche die Systeme und die von ihnen verwalteten und ausgetauschten Daten darstellt. Dann besprechen sie die Dokumente mit den Interviewpartnern und arbeiten das Feedback ein. Anhand dieser Dokumente bewerten die Berater die Anwendungslandschaft aus technischer Sicht und decken Probleme in der Kooperation der Systeme auf.

Parallel zur Erhebung der Anwendungslandschaft führt ein zweites Team von Beratern Interviews mit Vertretern der Fachabteilungen, um die Probleme mit der IT-Unterstützung zu eruieren. Die Ergebnisse stellt das Team dem Management als anonymes „Zufriedenheitsprofil“ zur Verfügung. Daraus geht hervor, mit welchen Aspekten der IT-Unterstützung die Nutzer zufrieden bzw. unzufrieden sind.

Anhand der technischen und der fachlichen Einschätzung bewerten die Berater die Anwendungslandschaft. Sie identifizieren technische und fachliche Unzulänglichkeiten und schlagen dem Management Maßnahmen in Form von Software-Aktualisierungen, Prozessanpassungen und Produktwechsell vor.

A.2 Fragebogen

Der folgende, zweiteilige Fragebogen wurde den Teilnehmern der Befragung als ausfüllbares PDF zugeschickt. Je Zeile war eine Antwortoption anzukreuzen. Fehlten in den ausgefüllten Fragebögen einzelne Antworten, wurden diese als Antwortoption „keine Angabe“ gewertet.

Die sieben Teilnehmer der Befragung waren in unterschiedlichen Rollen (Projektleiter, IT-Abteilungsleiter, Berater und Software Architekt) an Umgestaltungsprojekten beteiligt. Keiner der Teilnehmern hatte an den zuvor durchgeführten Experteninterviews teilgenommen (siehe Abschnitt A.1).

Der erste Teil des Fragebogens behandelt Aufgaben und Tätigkeiten in Umgestaltungsprojekten. Die Tabellen A.3–A.8 zeigt diesen Teil des Fragebogens inklusive der ausgewerteten Antworten.

Der zweite Teil (siehe Tabelle A.9) widmet sich Modellierungsproblemen. Diese wurden im Fragebogen als Anforderungen an einen Modellierungsansatz bezeichnet. Erst in der schriftlichen Ausarbeitung dieser Arbeit wurden die Anforderungen zu Modellierungsproblemen umformuliert. So wurde beispielsweise aus der Anforderung „widersprüchliche Informationen darstellen“ das Modellierungsproblem 3: „Bei der Modellierung können widersprüchliche Informationen auftreten“.

Von den acht als relevant bewerteten Anforderungen wurden nur sechs als Modellierungsprobleme in diese Arbeit übernommen. Die Anforderung „... zu erkennen, an welchen Stellen und in welchen Modellen eine Änderung Auswirkungen hat“ betrifft Modelle im Allgemeinen und ist nicht spezifisch für Umgestaltungsprojekte. Die Anforderung „... die Auswirkungen der Veränderungen transparent zu machen“ überschneidet sich mit mehreren anderen Anforderungen und fand daher ebenfalls nicht als Modellierungsproblem Eingang in diese Arbeit.

Für die Auswertung wurden die Antwortoptionen mit ganzzahligen Punkten versehen – vom minimal 0 Punkten für „keine Angabe“ bis maximal 4 Punkte für „stimme zu“. Je Antwortoption wurden die Punkte mit der Anzahl der Nennung multipliziert. Anschließend wurde das arithmetische Mittel gebildet. Die abgefragten Tätigkeiten und Probleme wurden als relevant gewertet, wenn das Mittel bei mindestens 2,5 Punkten lag, d.h. die Zustimmung im Schnitt über „stimme eher nicht zu“ lag.

Probleme identifizieren						
Zu Beginn eines Umgestaltungsprojekts muss noch nicht klar sein, was das Ziel der Umgestaltung darstellt. Die zu behebenden Probleme der Anwendungslandschaft zu identifizieren ist dann Teil des Projekts. Dabei kann nicht unbedingt auf aktuelle Modelle der Anwendungslandschaft und der von ihr unterstützten Prozesse zurückgegriffen werden. In diesem Schritt steht die Zielfindung für das Umgestaltungsprojekt im Vordergrund: Worin genau liegen die zu behebenden Probleme? Was ist zu tun? Der Schritt sollte daher folgende Aktivitäten beinhalten:	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊙ Ergebnis
Grobe Erfassung der eingesetzten Anwendungen						4,0
Erfassung der durch die Anwendungslandschaft unterstützten Prozesse						3,7
Sammeln und Bewerten von Schwachstellen der Anwendungslandschaft						3,2
Konsolidierung dieser Informationen zu einem Modell der Anwendungslandschaft						3,0
Bestimmen des Ziels der Umgestaltung						4,0
Insgesamt ist dieser Schritt relevant für Umgestaltungsprojekte						4,0
Ergänzen Sie gegebenenfalls weitere relevante Aktivitäten:						
Gab es in Ihren Umgestaltungsprojekten einen vergleichbaren Schritt?		nein			ja	

Tabelle A.3: Fragebogen zu Umgestaltungstätigkeiten, Abschnitt „Probleme identifizieren“

Strategie erarbeiten						
Dieser Schritt widmet sich der Frage, wie das Ziel der Umgestaltung erreicht werden kann. Dafür muss eine Reihe planerischer Aktivitäten ausgeführt werden. Dass es dabei um eine grobe Planung geht, spiegelt sich im Detailgrad der Ergebnisse wider. Der Schritt sollte daher folgende Aktivitäten beinhalten:	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊙ Ergebnis
Festlegung von abzulösenden und neu einzuführenden Softwaresystemen						3,4
Visualisieren der Ist- und Sollzustände der Anwendungslandschaft						3,7
Diskussion der zu erwartenden Veränderungen mit dem Management						3,8
Bestimmung eines groben Projektplans						4,0
Ausarbeitung und Kommunikation von strategischen Vorgaben für die Umgestaltung						3,8
Insgesamt ist dieser Schritt relevant für Umgestaltungsprojekte						3,7
Ergänzen Sie gegebenenfalls weitere relevante Aktivitäten:						
Gab es in Ihren Umgestaltungsprojekten einen vergleichbaren Schritt?		nein			ja	

Tabelle A.4: Fragebogen zu Umgestaltungstätigkeiten, Abschnitt „Strategie“

Handlungsbedarf ermitteln						
In diesem Schritt werden die grobe Planung und die strategischen Vorgaben auf einen konkreten Handlungsbedarf heruntergebrochen: Was bedeutet die Umgestaltung im Detail? Der Schritt sollte daher folgende Aktivitäten beinhalten:	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊖ Ergebnis
Handlungsbedarf für einzelne Anwendungen ermitteln						3,6
Detailanalyse von Ausschnitten der Anwendungslandschaft und der unterstützten Prozesse						3,2
Verfeinerung der Ist- und Soll-Modelle von Landschaft und Prozessen						3,3
Rückführung von Erkenntnissen in die strategische Planung						3,2
Insgesamt ist dieser Schritt relevant für Umgestaltungsprojekte						3,3
Ergänzen Sie gegebenenfalls weitere relevante Aktivitäten:						
Gab es in Ihren Umgestaltungsprojekten einen vergleichbaren Schritt?		nein				ja

Tabelle A.5: Fragebogen zu Umgestaltungstätigkeiten, Abschnitt „Handlungsbedarf“

Migrationsarchitektur planen						
	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊙ Ergebnis
Eine Migrationsarchitektur zeigt nicht nur, wann welche Änderungen in die produktive Anwendungslandschaft übernommen werden, sondern auch, wie die Prozesse zu diesem Zeitpunkt aussehen werden. Der Schritt sollte daher folgende Aktivitäten beinhalten:						
Umbaumaßnahmen in eine Reihenfolge bringen und in Phasen einteilen						3,6
Modelle der Anwendungslandschaft und der von ihr unterstützten Prozesse für jede Phase erstellen						3,1
Zu migrierende Daten bestimmen						3,7
Zeitpunkte für die Produktivstellung der Phasen bestimmen						2,7
Zustände und Veränderungen von Landschaft und Prozessen anhand der Modelle mit Fachabteilungen und IT abstimmen						3,3
Insgesamt ist dieser Schritt relevant für Umgestaltungsprojekte						3,7
Ergänzen Sie gegebenenfalls weitere relevante Aktivitäten:						
Gab es in Ihren Umgestaltungsprojekten einen vergleichbaren Schritt?		nein				ja

Tabelle A.6: Fragebogen zu Umgestaltungstätigkeiten, Abschnitt „Planen“

Umsetzung der Veränderungen						
	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊙ Ergebnis
Die Umsetzung der Planung findet Großteils in diesem Schritt statt. Der Schritt sollte daher folgende Aktivitäten beinhalten:						
Programmieren von Schnittstellen, Adaptern und Konvertern						3,6
Koordination von Projektmitarbeitern, die in Bezug auf eine Anwendung unterschiedliche Aufgaben haben						3,6
Den Projektmitarbeitern vermitteln, in welchem Kontext sie ihre Aufgaben erledigen						3,6
Unterschiedliche Sichtweisen darstellen und bei Missverständnissen vermitteln						3,3
Modelle bei Bedarf um detaillierte Informationen ergänzen						3,0
Insgesamt ist dieser Schritt relevant für Umgestaltungsprojekte						4,0
Ergänzen Sie gegebenenfalls weitere relevante Aktivitäten:						
Gab es in Ihren Umgestaltungsprojekten einen vergleichbaren Schritt?		nein			ja	

Tabelle A.7: Fragebogen zu Umgestaltungstätigkeiten, Abschnitt „Umsetzung“

Testen der Veränderungen						
	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊙ Ergebnis
In diesem Schritt werden die Veränderungen der Anwendungen und Prozesse getestet. Der Schritt sollte daher folgende Aktivitäten beinhalten:						
Testpläne aufstellen, die auf die Migrationsarchitektur abgestimmt sind						3,1
Rahmenbedingungen für Tests und die Veränderungen an den Prozessen an die Tester kommunizieren						3,1
Anwendungen einzeln testen						3,3
Fachliche Prozesse anwendungsübergreifend testen						3,9
Insgesamt ist dieser Schritt relevant für Umgestaltungsprojekte						3,7
Ergänzen Sie gegebenenfalls weitere relevante Aktivitäten:						
Gab es in Ihren Umgestaltungsprojekten einen vergleichbaren Schritt?		nein				ja

Tabelle A.8: Fragebogen zu Umgestaltungstätigkeiten, Abschnitt „Testen“

Der Modellierungsansatz soll dabei helfen:	0 keine Angabe	1 stimme nicht zu	2 stimme eher nicht zu	3 stimme eher zu	4 stimme zu	⊖ Ergebnis
Die Informationsmenge handhabbar zu machen						3,7
Den Zweck von Modellen nachvollziehbar zu machen						2,1
Die Sichtweise der an der Modellierung Beteiligten nachvollziehbar zu machen						2,4
Widersprüchliche Informationen darzustellen						2,9
Widersprüche zu verhandeln und möglichst zu beseitigen						2,0
Abhängigkeiten in abteilungsübergreifenden und anwendungsübergreifenden Prozessen zu erkennen						3,7
Indirekte und rein fachliche Abhängigkeiten zwischen Anwendungen abzubilden						3,4
Den Beteiligten zu verdeutlichen, wie die Anwendungslandschaft von ihren Anwendern verwendet wird						2,1
Den Beteiligten zu verdeutlichen, wie die Anwendungslandschaft die Geschäftsprozesse unterstützt						2,9
Veränderungen einer Anwendungslandschaft über die Zeit darzustellen						3,4
Die Auswirkungen der Veränderungen transparent zu machen						3,1
Zu erkennen, an welchen Stellen und in welchen Modellen eine Änderung Auswirkungen hat						3,6

Tabelle A.9: Fragebogen zu Modellierungsproblemen

Anhang B

Bewertung ausgewählter Modellierungsansätze

Im Rahmen der Bewertung von Modellierungsansätzen in Kapitel 4 wurde eine beispielhafte Modellierungsaufgabe mit den Ansätzen gelöst. Die Modellierungsaufgabe wurde so gewählt, dass sie zu den zu bewertenden Anforderungen passt:

- Sie ist an ein reales Umgestaltungsprojekt angelehnt und steht damit beispielhaft für eine praxisrelevante Modellierungsaufgabe.
- Sie erfordert Modelle von Geschäftsprozessen und ihrer IT-Unterstützung durch eine Anwendungslandschaft.
- Sie erfordert Modelle eines Ausgangszustands und eines daraus abgeleiteten Soll-Zustands, sodass exploratives Modellieren und Modellmanipulationen anhand des Beispiels gezeigt werden können.

Die Modellierungsaufgabe handelt von der fiktiven Pfefferminzia-Versicherung, die schon in Kapitel 3 und Kapitel 6 als Beispiel diente. Die folgende Beschreibung des Beispiels ist eine Wiederholung aus Abschnitt 3.5:

Die Pfefferminzia Versicherung will die Bearbeitungszeit von Vorgängen wie z.B. der Schadensregulierung verkürzen und zudem die Auskunftsfähigkeit über den Status der Vorgänge erhöhen. Erreicht werden sollen diese Ziele durch ein Umgestaltungsprojekt, mit dem die Anwendungslandschaft um ein Workflow-System ergänzt wird. Dieses soll die IT-Unterstützung der Vorgangsbearbeitung verbessern.

An einem Ausschnitt des Prozesses „Leistungsfall bearbeiten“ werden der Ist-Zustand der Anwendungslandschaft und ihr angestrebter Soll-Zustand vorgestellt:

In der Abteilung *Posteingang* werden per Post eingegangene, papierhafte Dokumente gescannt und im System *Dokumentverwaltung* digital erfasst. Anhand des Nachnamens der versicherten Person benachrichtigt die Dokumentverwaltung per Mail die für den betroffenen Buchstabenkreis zuständige Gruppe von Sachbearbeitern. Der nächste freie Sachbearbeiter beginnt mit der Bearbeitung des Falls und legt dazu mit *Microsoft Excel* eine Vorgangsmappe für den Fall an. Anhand der Dokumente zum Leistungsfall in der Dokumentverwaltung und den im *Leistungssystem* hinterlegten Leistungsdaten der versicherten Person wird die zu erbringende Leistung geprüft. Das Ergebnis der Prüfung wird in der Vorgangsmappe dokumentiert. Falls die zu erbringenden Leistungen nicht restlos geklärt werden können, gibt der Sachbearbeiter die Vorgangsmappe weiter an einen Leistungsspezialisten. Dieser prüft den unklaren Leistungsfall anhand der Dokumente in der Dokumentverwaltung. Anschließend dokumentiert er sein Ergebnis in der Vorgangsmappe und gibt diese zurück an den Sachbearbeiter.

Zukünftig sollen die Dokumenten eines Leistungsfalls nicht nur digitalisiert werden, sondern dabei automatisch ein Vorgang im neuen Workflow-System angelegt werden. Das Workflow-System prüft selbstständig die zu erbringende Leistung und dokumentiert das Ergebnis in der Vorgangsmappe. Falls der Leistungsfall nicht restlos geklärt werden kann, informiert das Workflow-System per Mail einen Leistungsspezialisten. Dieser prüft den unklaren Leistungsfall anhand der Dokumente in der Dokumentverwaltung. Anschließend dokumentiert er sein Ergebnis in der Vorgangsmappe und signalisiert dem Workflow-System damit, dass es nun die Bearbeitung fortführen kann.

In den folgenden Unterabschnitten wird dieses Beispiel mit den bewerteten Modellierungsansätzen modelliert, um die Ergebnisse der Bewertung zu illustrieren.

B.1 ArchiMate

Als Quellen für die Bewertung dienen die Spezifikationsdokumente der Open Group ([The13a], [The13b], [The13c] und [The13d]). Tabelle B.1 zeigt die Ergebnisse. Die Beispielmodelle wurde mit dem Werkzeug *BiZZdesign Architect 4.5.2* [Biz] erstellt.

- Modelle des Ist-Zustands der Pfefferminzia Anwendungslandschaft: Abbildung B.1, Abbildung B.2
- Modelle des Soll-Zustands der Pfefferminzia Anwendungslandschaft: Abbildung B.3, Abbildung B.4

Abbildung B.5 zeigt eine Legende aller verwendeten Modellelementtypen und Relationen.

Anforderung 1 (IT-gestützte Arbeitsgegenstände)	Bewertung: ●
--	--------------

Arbeitsgegenstände und -ergebnisse werden als „Business Objects“ modelliert. Ob diese IT-gestützt sind, lässt sich grafisch nur durch Relationen zu anderen Modellelementen darstellen: „Business Objects“ können von „Data Objects“ realisiert werden, welche wiederum mit „Application Components“ in Beziehung stehen können. Eine direkte Beziehung zwischen Arbeitsgegenständen und Softwaresystemen ist nur über eine unspezifische „Association“-Relation umsetzbar.

Anforderung 2 (Manuelle und IT-gestützte Tätigkeiten)	Bewertung: ●
--	--------------

Tätigkeiten werden als „Business Function“ oder „Business Interaction“ modelliert und können einer „Business Role“ (und damit einem menschlichen Akteur) zugewiesen werden. Über eine „used by“-Beziehung können die Tätigkeiten mit „Application Services“ in Beziehung gesetzt werden (siehe *Application Usage Viewpoint* [The13c, S.14]). Eine direkte Beziehung zwischen Tätigkeiten und einem Softwaresystem ist nur über eine unspezifische „Association“-Relation umsetzbar.

Anforderung 3 (Koordination)	Bewertung: ●
-------------------------------------	--------------

Mit dem Viewpoint *Actor Cooperation* [The13c, S.6] kann ein Überblick über die Kommunikationswege zwischen Akteuren dargestellt werden. Jedoch gibt es in dieser Darstellung keinen Bezug zu Arbeitsabläufen, da diese mit dem Viewpoint *Business Process Cooperation* [The13c, S.6] dargestellt werden.

Anforderung 4 (Fachliche Konzepte)	Bewertung: +
Der Business Layer bietet Darstellungsmöglichkeiten für fachliche Konzepte. Dies veranschaulicht der <i>Information Structure Viewpoint</i> [The13c, S.18].	
Anforderung 5 (Fachliche Architektur)	Bewertung: +
Einen vordefinierten Viewpoint für fachliche Architekturen gibt es zwar nicht, doch können Modellierer den <i>Application Cooperation Viewpoint</i> [The13c, S.12] und den <i>Application Structure Viewpoint</i> [The13c, S.13] kombinieren und mit fachlichen Begriffen arbeiten, um eine fachliche Architektur zu modellieren.	
Anforderung 6 (Technologische Sichten)	Bewertung: +
Der Application Layer bietet Darstellungsmöglichkeiten für technologische Sichten, z.B. <i>Application Cooperation Viewpoint</i> [The13c, S.12] und <i>Application Structure Viewpoint</i> [The13c, S.13].	
Anforderung 7 (Beziehungen zwischen Sichten)	Bewertung: +
Fachliche und technologische Sichten können miteinander verbunden und sogar im selben Modell dargestellt werden. Einige der Viewpoints zeigen sowohl fachliche als auch technische Aspekte.	
Anforderung 8 (Zweifel an Brauchbarkeit)	Bewertung: –
Es gibt keine Sprachmittel, um Zweifel an der Brauchbarkeit eines Modells auszudrücken.	
Anforderung 9 (Validierung im Modellierungsprozess)	Bewertung: –
ArchiMate beschreibt kein Modellierungsvorgehen.	
Anforderung 10 (Vages Modellieren)	Bewertung: –
Die subjektive Brauchbarkeit konzeptueller Modelle wird nicht thematisiert.	
Anforderung 11 (Einsatzszenarien)	Bewertung: +
Die Sammlung von Viewpoints in [The13c] gibt einen Überblick, welche Einsatzszenarien von ArchiMate unterstützt werden. Für jeden Viewpoint werden die adressierten Stakeholder, das Abstraktionsniveau, der Zweck und die mit dem Viewpoint beantwortbaren Fragestellungen beschrieben. Mehrere der Viewpoints wurden verwendet, um die Modellierungsaufgabe zu lösen.	
Anforderung 12 (Modellmanipulation und Analyse)	Bewertung: –
ArchiMate beschreibt keine Modellmanipulationen und Modellanalysen.	
Anforderung 13 (Status der Modelle)	Bewertung: –

Modellstatus sind nicht Teil der Spezifikation und müssten in einem Modellierungswerkzeug oder durch eine Spracherweiterung (vgl. „language extension mechanisms“ [The13a]) ergänzt werden.

Tabelle B.1: Bewertung ArchiMate

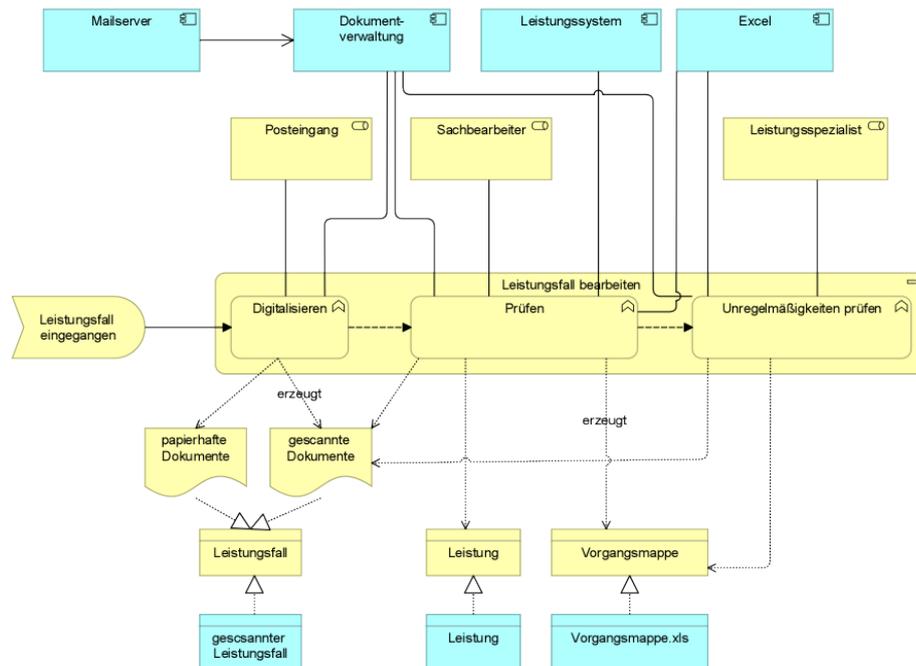


Abbildung B.1: Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als Kombination der ArchiMate Viewpoints *Business Process*, *Application Usage* und *Information Infrastructure*

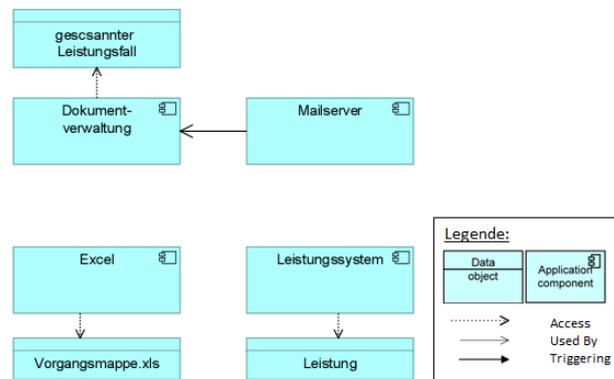


Abbildung B.2: Fachliche Architektur (Ist) durch Kombination der ArchiMate Viewpoints *Application Cooperation* und *Application Structure*

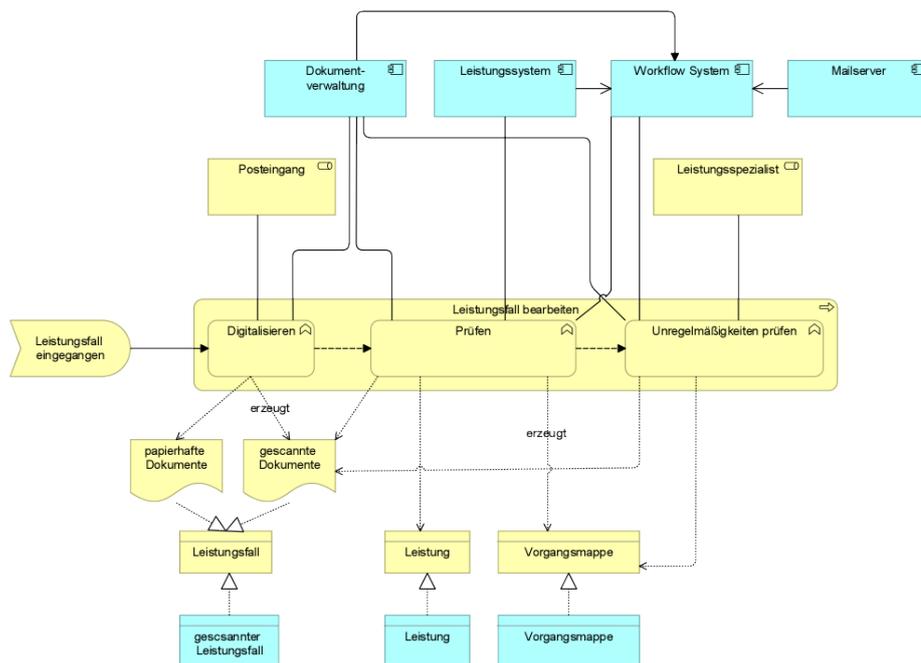


Abbildung B.3: Geschäftsprozess „Leistungsfall bearbeiten“ (Soll) als Kombination der ArchiMate Viewpoints *Business Process*, *Application Usage* und *Information Infrastructure*

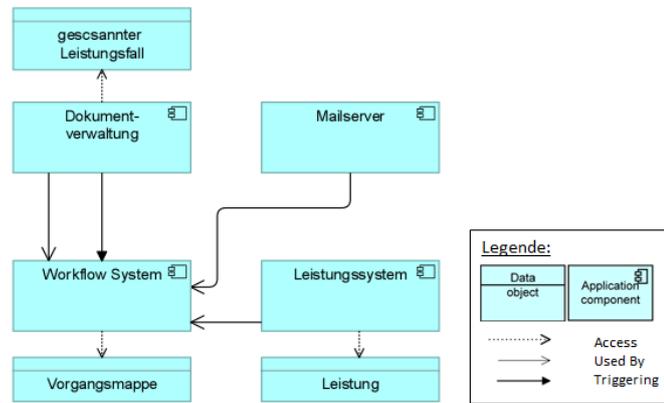


Abbildung B.4: Fachliche Architektur (Soll) durch Kombination der ArchiMate Viewpoints *Application Cooperation* und *Application Structure*

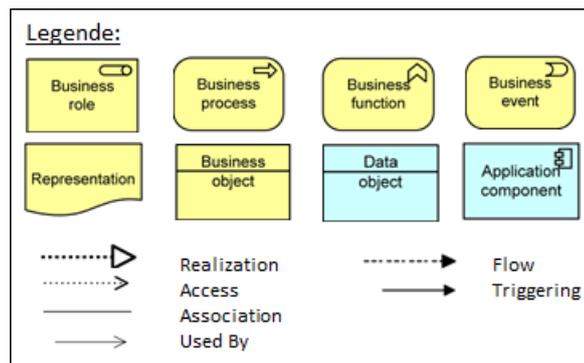


Abbildung B.5: Legende aller verwendeten ArchiMate Modellelementtypen und Relationen

B.2 ARIS

Als Quellen für die Bewertung dienen [KNS92], [Sch02] und [Leh08]. Tabelle B.2 zeigt die Ergebnisse. Die Beispiele wurde mit dem Werkzeug *ARIS Express 2.4b* erstellt.

- Modelle des Ist-Zustands der Pfefferminzia Anwendungslandschaft: B.6, B.7
- Modelle des Soll-Zustands der Pfefferminzia Anwendungslandschaft: B.8, B.9

Die Abbildungen B.10 und B.11 zeigen Legenden aller verwendeten Modellelementtypen und Relationen.

Anforderung 1 (IT-gestützte Arbeitsgegenstände)	Bewertung: –
Arbeitsgegenstände und -ergebnisse werden als „Umfelddaten“, „Nachricht“ oder „Leistung“ modelliert und können nicht direkt mit Softwaresystemen in Beziehung gesetzt werden.	
Anforderung 2 (Manuelle und IT-gestützte Tätigkeiten)	Bewertung: ●
Tätigkeiten werden als „Funktionen“ modelliert, die mit „menschlicher Arbeitsleistung“ und „Anwendungssoftware“ in Verbindung stehen können. ARIS gibt jedoch nicht vor, dass Funktionen unterschiedlich dargestellt werden sollen, wenn sie manuell oder IT-gestützt sind.	
Anforderung 3 (Koordination)	Bewertung: –
Es gibt keine Sprachmittel, um die Koordination zwischen Akteuren zu veranschaulichen.	
Anforderung 4 (Fachliche Konzepte)	Bewertung: +
Die Datensicht stellt fachliche Konzepte dar.	
Anforderung 5 (Fachliche Architektur)	Bewertung: –
Keine der Sichten eignet sich, um fachliche Architekturen darzustellen.	
Anforderung 6 (Technologische Sichten)	Bewertung: –
Keine der Sichten des ARIS-Hauses ist auf Ebene des DV-Konzepts definiert.	
Anforderung 7 (Beziehungen zwischen Sichten)	Bewertung: ●
Die fünf ARIS-Sichten sind in einander integriert. Allerdings deckt die Definition der Sichten auf Fachkonzept-Ebene keine technischen Sichten ab. Eigene Sichten zu definieren ist in ARIS nicht vorgesehen.	
Anforderung 8 (Zweifel an Brauchbarkeit)	Bewertung: –

Es gibt keine Sprachmittel, um Zweifel an der Brauchbarkeit eines Modells auszudrücken.

Anforderung 9 (Validierung im Modellierungsprozess) Bewertung: –

Das ARIS Vorgehensmodell zur Prozessmodellierung nach Keller et al. ([KNS92]) sieht eine Validierung von Modellen vor, ohne näher darauf einzugehen. In [Sch02] wird Simulation als Mittel zur Validierung vorgeschlagen. Lehmann ergänzt das Vorgehensmodell in [Leh08] um Rollen in der Modellierung. Fachexperten nehmen keine Rolle ein, da sie ihr Wissen bereits in „Prozessbeschreibungen“ eingebracht haben, welche als Basis der Modelle dienen.

Anforderung 10 (Vages Modellieren) Bewertung: ●

Dass Modelle vage sind, wird zwar thematisiert, aber kein Umgang damit vorgeschlagen: „[...] bei der isolierten Entwicklung [mehrerer Modelle] für eine Untersuchungseinheit [...] ist zu erwarten, daß [...] Unstimmigkeiten der Modelle aufgezeigt werden. [KNS92, S.19f]

Anforderung 11 (Einsatzszenarien) Bewertung: ●

ARIS wurde entwickelt, um Geschäftsprozesse und ihre IT-Unterstützung zu gestalten. Für dieses Szenario wird eine methodische Unterstützung in Form eines Vorgehensmodells beschrieben (vgl. [KNS92, S.17ff]), das jedoch kein Modellierungsvorgehen für die Sichten einschließt. Weitere Einsatzszenarien umfassen Aufgaben des *Geschäftsprozessmanagements* wie Prozessbewertung, Simulation und Wissensmanagement (vgl. [KNS92]).

Anforderung 12 (Modellmanipulation und Analyse) Bewertung: ●

In ARIS dienen Modelle zur Ist-Analyse und zur Entwicklung von Soll-Zuständen. Konkrete Modellmanipulationen werden jedoch nicht beschrieben. Eine Möglichkeit der Analyse ist die werkzeuggestützte Simulation von EPKs.

Anforderung 13 (Status der Modelle) Bewertung: ●

In [KNS92, S.86f] wird ein Metamodell zur Verwaltung von Modellversionen definiert, in der jedoch kein Status enthalten ist. Keller et al. thematisieren darüber hinaus die Notwendigkeit von Modellvarianten in Gestaltungsprozessen. Eine konkrete Umsetzung schlagen sie jedoch nicht vor.

Tabelle B.2: Bewertung ARIS

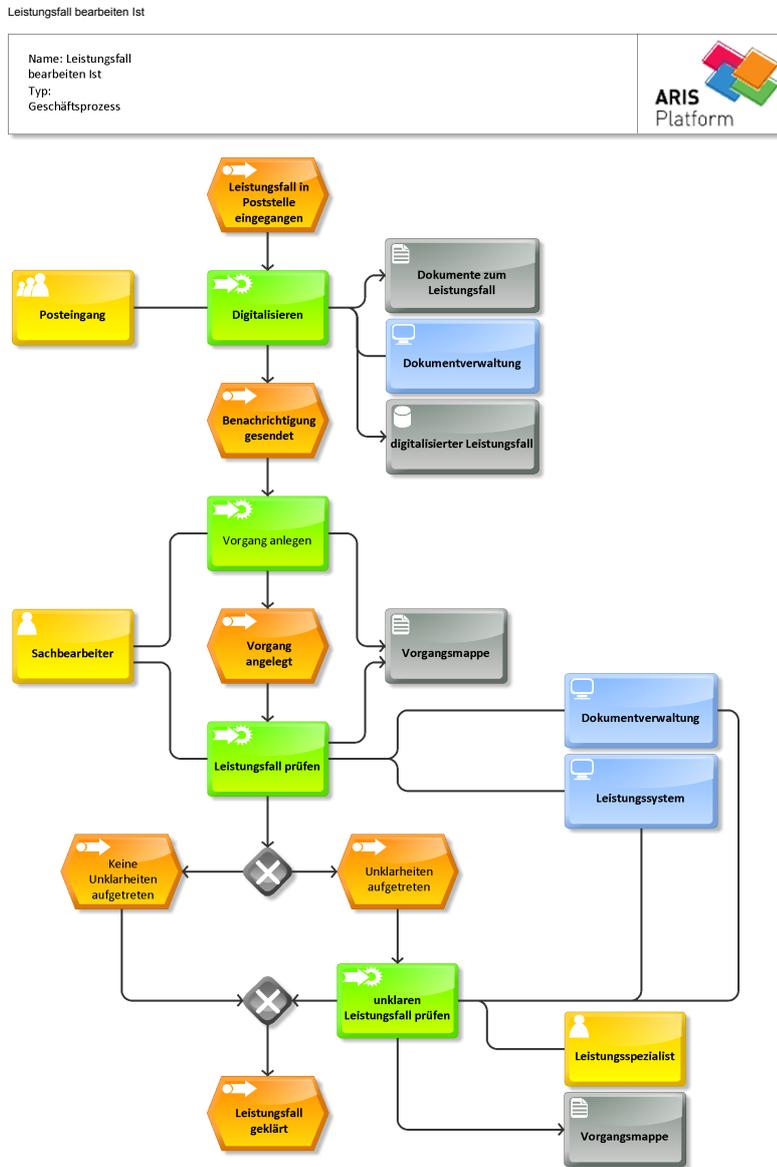


Abbildung B.6: Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als ARIS EPK



Abbildung B.7: Fachliche Architektur (Ist) als ARIS Systemlandschaft

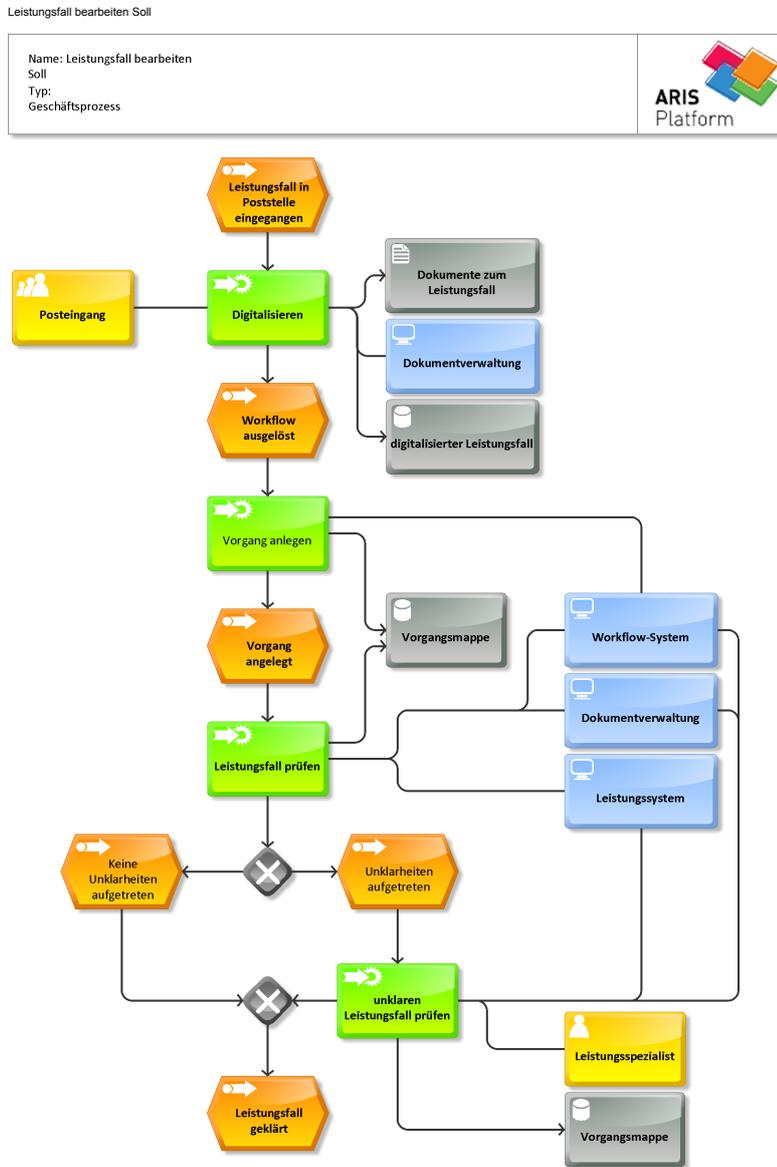


Abbildung B.8: Geschäftsprozess „Leistungsfall bearbeiten“ (Soll) als ARIS EPK

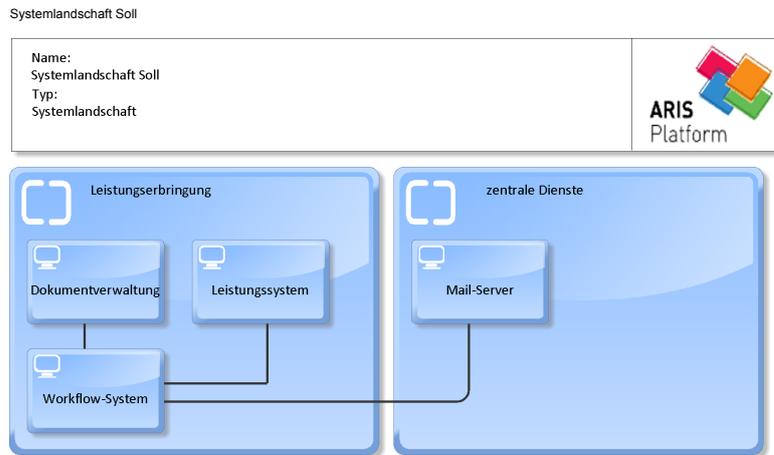


Abbildung B.9: Fachliche Architektur (Soll) als ARIS Systemlandschaft



Abbildung B.10: Legende zur ARIS EPK

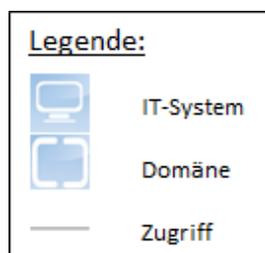


Abbildung B.11: Legende zur ARIS Systemlandschaft

B.3 BEN

Als Quelle für die Bewertung dient [Win10]. Tabelle B.3 zeigt die Ergebnisse. Die Beispiele wurden mit dem Werkzeug *ADOben 1.0* erstellt. Da ADOben nicht über Modelltypen zur Darstellung von Geschäftsprozessen verfügt, wurde nur die fachliche Architektur der Anwendungslandschaft modelliert.

- Modelle des Ist-Zustands der Pfefferminzia Anwendungslandschaft: B.12
- Modelle des Soll-Zustands der Pfefferminzia Anwendungslandschaft: B.13
- Weitere Modelle: B.15 , B.16

Abbildung B.14 zeigt eine Legende der ADOben Applikationslandschaft.

Anforderung 1 (IT-gestützte Arbeitsgegenstände)	Bewertung: –
Arbeitsgegenstände und -ergebnisse werden in BEN als „Geschäftsobjekt“ modelliert. Ein Geschäftsobjekt kann nicht direkt mit Softwaresystemen in Beziehung gesetzt werden.	
Anforderung 2 (Manuelle und IT-gestützte Tätigkeit)	Bewertung: •
Manuelle und IT-gestützte Tätigkeiten werden in BEN als „Aufgabe“ modelliert. Eine Aufgabe wird von einer „Rolle“ ausgeführt und kann durch einen IT-basierten, fachlichen Service unterstützt werden. BEN legt keine konkrete Notation fest, die manuelle und IT-gestützte Aufgaben unterscheidet.	
Anforderung 3 (Koordination)	Bewertung: –
Es gibt keine Sprachmittel, um die Koordination zwischen Akteuren zu veranschaulichen.	
Anforderung 4 (Fachliche Konzepte)	Bewertung: +
Mit dem Modelltyp „Informationsmodell“ können fachliche Konzepte dargestellt werden.	
Anforderung 5 (Fachliche Architektur)	Bewertung: +
Fachliche Architekturen können mit dem Modelltyp <i>Applikationslandschaft</i> dargestellt werden.	
Anforderung 6 (Technologische Sichten)	Bewertung: –
Es gibt keine geeigneten Modelltypen.	
Anforderung 7 (Beziehung zwischen Sichten)	Bewertung: +

BEN beschreibt sowohl fachliche als auch technische Modelltypen, die über das Metamodell miteinander verbunden sind.

Anforderung 8 (Zweifel an Brauchbarkeit)	Bewertung: –
---	--------------

Es gibt keine Sprachmittel, um Zweifel an der Brauchbarkeit eines Modells auszudrücken.

Anforderung 9 (Validierung im Modellierungsprozess)	Bewertung: –
--	--------------

BEN umfasst kein Vorgehen für die Modellierung.

Anforderung 10 (Vages Modellieren)	Bewertung: –
---	--------------

Konzeptuelle Modellierung wird im Rahmen der *Informationsmodellierung* erwähnt. Dabei sollen z.B. „relevante Aussagen über die Gegenstände und Sachzusammenhänge des Realitätsbereichs“ [Win10, S.135f] aufgenommen werden. Dass diese jedoch nur vage modelliert werden können, wird im BEN nicht thematisiert.

Anforderung 11 (Einsatzszenarien)	Bewertung: +
--	--------------

BEN ist in vier „grundsätzlichen Projekttypen“ [Win10, S.71] anwendbar, etwa der *fachliche getriebenen Innovation*. Projekte dieser vier Typen können als Umgestaltungsprojekte klassifiziert werden. BEN beschreibt für jeden der Typen, in welcher Reihenfolge und für welchen Zweck Modelle erstellt und analysiert werden sollen.

Anforderung 12 (Modellmanipulation und Analyse)	Bewertung: –
--	--------------

BEN geht nicht darauf ein, wie Modelle manipuliert und analysiert werden.

Anforderung 13 (Status der Modelle)	Bewertung: –
--	--------------

In BEN werden keine Angaben zum Gestaltungsprozess von Modellen und zu ihrem Status gemacht.

Tabelle B.3: Bewertung BEN

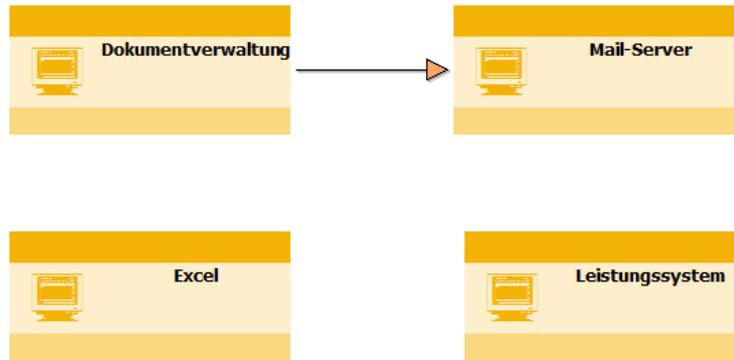


Abbildung B.12: Fachliche Architektur (Ist) als ADOben Applikationslandkarte

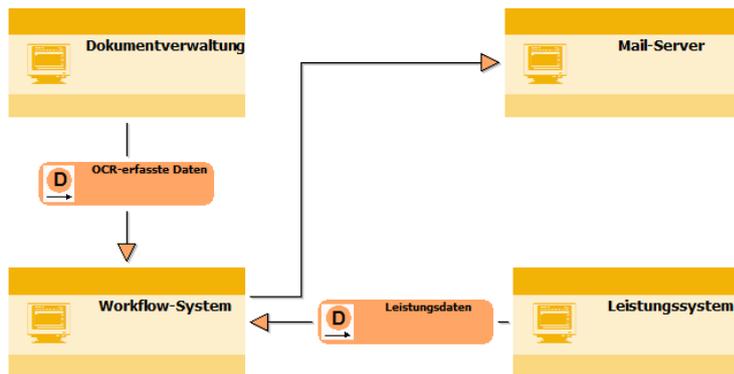


Abbildung B.13: Fachliche Architektur (Soll) als ADOben Applikationslandkarte

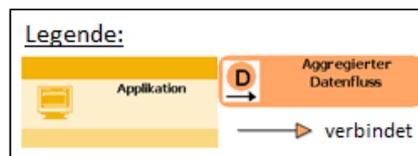


Abbildung B.14: Legende zur ADOben Applikationslandkarte



Abbildung B.15: Modell vom Typ ADOben *Applikationen (Bestandsführung)* – wird von den Modellen B.12 und B.13 referenziert



Abbildung B.16: Modell vom Typ ADOben *Datenmodell* – wird vom Modell B.15 referenziert

B.4 EAM Patterns

Als Quelle für die Bewertung dient der EAM Pattern Catalog 1.0 ([BELM08]). Tabelle B.4 zeigt die Ergebnisse. Die Beispiele wurden mit *Microsoft PowerPoint 2010* erstellt.

- Modelle des Ist-Zustands der Pfefferminzia Anwendungslandschaft: B.19, B.21
- Modelle des Soll-Zustands der Pfefferminzia Anwendungslandschaft: B.20, B.22

Die Abbildungen B.17 und B.18 zeigen Legenden zu den verwendeten Viewpoints.

Anforderung 1 (IT-gestützte Arbeitsgegenstände)	Bewertung: ●
--	--------------

Viewpoint V-51 zeigt, wie Arbeitsgegenstände und -ergebnisse als „Business Objects“ in einem IT-gestützten Geschäftsprozess verwendet werden. Wie menschliche Akteure in einem Geschäftsprozess Arbeitsgegenstände und -ergebnisse handhaben, kann mit diesem Viewpoint nicht dargestellt werden. Damit eignet sich der Viewpoint nur eingeschränkt, um Veränderungen der IT-Unterstützung zwischen Ist- und Sollprozessen darzustellen.

Anforderung 2 (Manuelle und IT-gestützte Tätigkeiten)	Bewertung: –
--	--------------

In den Viewpoints V-18 und V-51 können die in einem Geschäftsprozess verwendeten „Services“ dargestellt werden. Ein Service kann durch eine „Business Application“ realisiert werden. Eine direkte Beziehung zwischen Tätigkeiten und einem Softwaresystem ist jedoch nicht umsetzbar.

Anforderung 3 (Koordination)	Bewertung: ●
-------------------------------------	--------------

Mit dem Viewpoint V-52 kann ein Überblick über die Kommunikationswege zwischen Akteuren dargestellt werden. Jedoch gibt es in einer solchen Darstellung keinen Bezug zu Arbeitsabläufen, da diese mit dem Viewpoint V-51 dargestellt werden.

Anforderung 4 (Fachliche Konzepte)	Bewertung: ●
---	--------------

Fachliche Konzepte können eingeschränkt mit Viewpoint V-47 dargestellt werden.

Anforderung 5 (Fachliche Architektur)	Bewertung: +
--	--------------

Fachliche Architekturen können mit Viewpoint V-48 umgesetzt werden.

Anforderung 6 (Technologische Sichten)	Bewertung: +
Viewpoint V-66 und weitere, aus der UML entlehnte Viewpoints eignen sich zur Darstellung technischer Sichten.	
Anforderung 7 (Beziehung zwischen Sichten)	Bewertung: ●
Einige der Viewpoints stehen miteinander in Beziehung (z.B. als Spezialisierung). Außerdem besteht mit den <i>Information Patterns</i> eine Basis für ein Sichtenübergreifendes Informationsmodell. Konkrete Verbindungen zwischen fachlichen und technischen Sichten werden jedoch nicht im Katalog erwähnt.	
Anforderung 8 (Zweifel an Brauchbarkeit)	Bewertung: –
Es gibt keine Sprachmittel, um Zweifel an der Brauchbarkeit eines Modells auszudrücken.	
Anforderung 9 (Validierung im Modellierungsprozess)	Bewertung: –
Vorgehensweisen für die Modellierung werden in den EAM Patterns nicht berücksichtigt, da die Modelle als Visualisierung zuvor gesammelter Informationen aufgefasst werden. Um diese Anforderungen zu erfüllen, müsste der Ansatz um passende „Informationserfassungsmuster“ erweitert werden.	
Anforderung 10 (Vages Modellieren)	Bewertung: –
Siehe Bewertung von Anforderung 9.	
Anforderung 11 (Einsatzszenarien)	Bewertung: +
Mehrere der <i>Concerns</i> betreffen Fragestellungen, die für Umgestaltungsprojekte relevant sind: Z.B. C-9, C-34, C-35, C-36, C-78	
Anforderung 12 (Modellmanipulation und Analyse)	Bewertung: +
Die Vorgehensmuster beschreiben, wie eine Anwendungslandschaft mit Hilfe von Darstellungsmustern zu analysieren ist, um bestimmte Fragestellungen zu beantworten. Für Umgestaltungsprojekte sind u.a. folgende Vorgehensmuster relevant: M-14, M-15, M-29, M-30	
Anforderung 13 (Status der Modelle)	Bewertung: –
Die Darstellungen im EAM-Pattern-Katalog wurden als <i>Visualisierungen</i> eines zugrundeliegenden Informationsmodells konzipiert, nicht als zu gestaltende, grafische Modelle. Weder auf den Status der Visualisierungen noch auf den des Informationsmodells wird im Katalog eingegangen.	

Tabelle B.4: Bewertung EAM Pattern Catalog

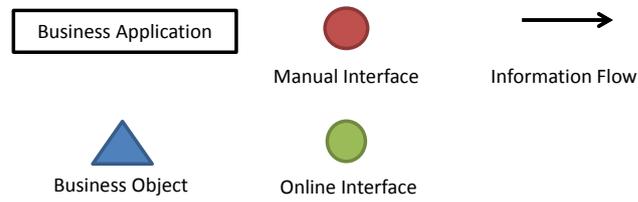


Abbildung B.17: Legende zum EAM-Pattern V-48

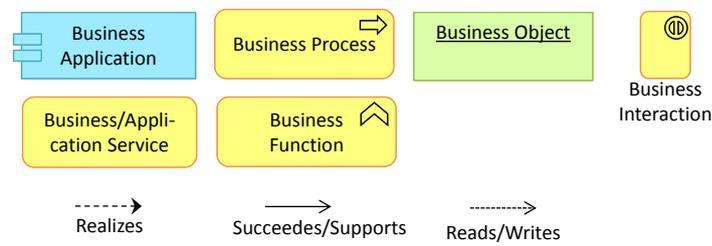


Abbildung B.18: Legende zu den EAM-Patterns V-18 und V-51

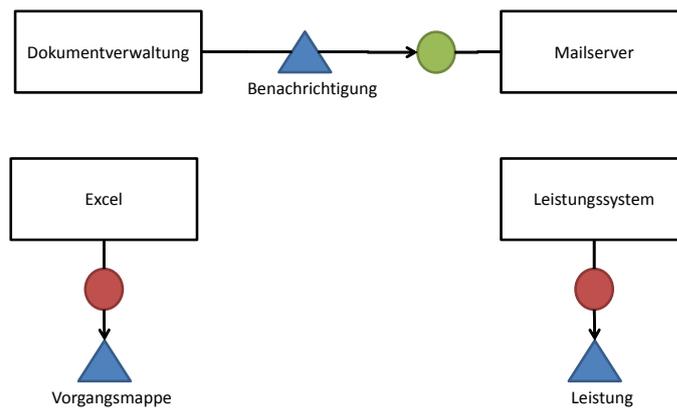


Abbildung B.19: Fachliche Architektur (Ist) als EAM-Pattern V-48

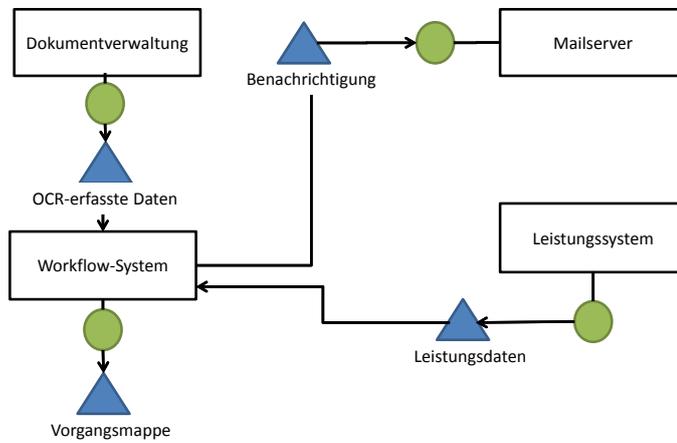


Abbildung B.20: Fachliche Architektur (Soll) als EAM-Pattern V-48

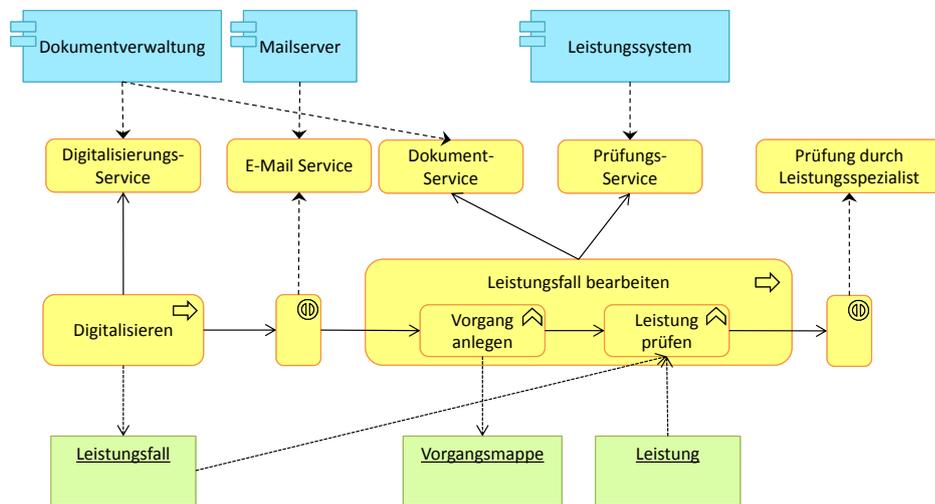


Abbildung B.21: Ausschnitt aus Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als Kombination der EAM-Pattern V-18 und V-51

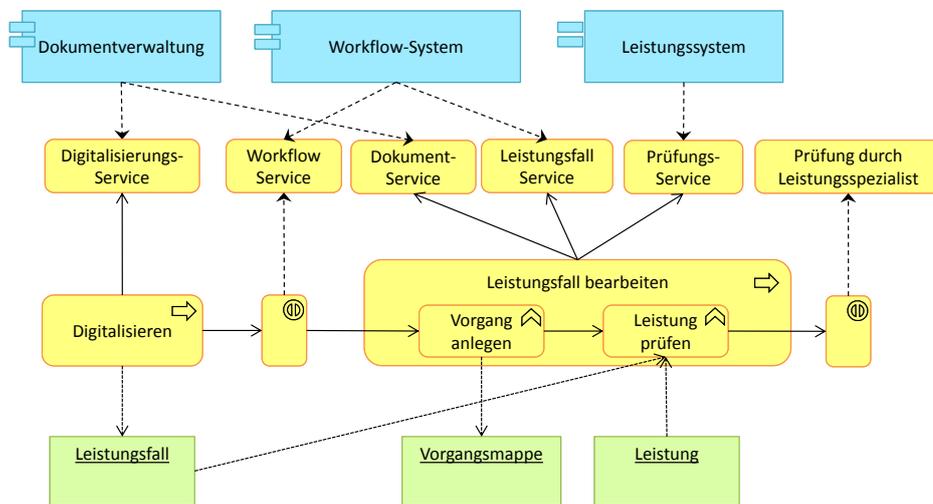


Abbildung B.22: Ausschnitt aus Geschäftsprozess „Leistungsfall bearbeiten“ (Soll) als Kombination der EAM-Pattern V-18 und V-51

B.5 eGPM

Als Quellen für die Bewertung dienen [BKS06], [BH12] und [WPS14]. Tabelle B.5 zeigt die Ergebnisse. Die Beispielm Modelle wurde mit dem Werkzeug *eGPM Anwendungsbibliothek 2014.10.08 für ADONIS 5.0* erstellt.

- Modelle des Ist-Zustands der Pfefferminzia Anwendungslandschaft: B.23, B.24
- Modelle des Soll-Zustands der Pfefferminzia Anwendungslandschaft: B.27, B.28
- Weitere Modelle: B.29

Die Abbildungen B.25, B.26 und B.30 zeigen Legenden zu den Modelltypen.

Anforderung 1 (IT-gestützte Arbeitsgegenstände)	Bewertung: +
Arbeitsgegenstände und -ergebnisse werden als „Gegenstand“ modelliert und sind über ein Attribut „IT-Unterstützung“ mit den Systemen einer IT-Landschaft verknüpfbar.	
Anforderung 2 (Manuelle und IT-gestützte Tätigkeiten)	Bewertung: +
In Kooperationsbildern gehen Tätigkeiten immer von menschlichen oder IT-basierten Akteuren aus. Akteure können IT-gestützte „Gegenstände“ verwenden und erzeugen. Solche Gegenstände werden grafisch mit einem Blitz-Symbol annotiert.	
Anforderung 3 (Koordination)	Bewertung: +
Kooperationsbilder machen durch die Relationen „informiert mit“ und „gibt Gegenstand weiter“ explizit, wie Akteure ihre Arbeitsabläufe koordinieren.	
Anforderung 4 (Fachliche Konzepte)	Bewertung: +
Mit den Modelltyp „Begriffsmodell“ können Geschäftsprozesse und fachliche Konzepte dargestellt werden.	
Anforderung 5 (Fachliche Architektur)	Bewertung: +
IT-Landschaften abstrahieren von technischen Details und sind mit fachlichen Begriffen verknüpfbar.	
Anforderung 6 (Technologische Sichten)	Bewertung: -
Die eGPM umfasst keine Modelltypen für technologische Sichten.	
Anforderung 7 (Beziehung zwischen Sichten)	Bewertung: +

Verknüpfungen zwischen verschiedenen Modellen sind möglich und können durch werkzeuggestützte Abfragen ausgewertet werden. Modellelemente eines Kooperationsbildes und einer IT-Landschaft können dasselbe Modellelement eines Begriffsmodells referenzieren.

Anforderung 8 (Zweifel an Brauchbarkeit)	Bewertung: +
---	--------------

Die Modellelementtypen des Kooperationsbildes und der IT-Landschaft verfügen über ein auswertbares, textuelles Attribut „offene Fragen“. Ist dieses Attribut an einem Modellelement gesetzt, wird dessen Darstellung um ein ?-Symbol ergänzt. Dadurch können Zweifel an der Brauchbarkeit eines Modells ausgedrückt werden.

Anforderung 9 (Validierung im Modellierungsprozess)	Bewertung: +
--	--------------

In Workshops unter Leitung eines Modellierers und Moderators modellieren die betroffenen Akteure gemeinschaftlich und können unmittelbar Rückmeldung geben.

Anforderung 10 (Vages Modellieren)	Bewertung: +
---	--------------

Kooperationsbilder stellen exemplarische Prozessverläufe als Szenarien dar (vgl. [Hof11]). Geschäftsprozesse werden nicht „vollständig“ modelliert, sondern nur die für den Modellzweck relevante Ausschnitte.

Anforderung 11 (Einsatzszenarien)	Bewertung: ●
--	--------------

Mehrere Einsatzszenarien werden in Stichpunkten beschrieben, unter anderem die Umgestaltung von Anwendungslandschaften. Detailliert wird auf den Umgang mit Modellen nur im Szenario „Objektorientierter Entwurf“ (siehe [BKS06]) eingegangen.

Anforderung 12 (Modellmanipulation und Analyse)	Bewertung: ●
--	--------------

Modellmanipulationen, wie die Ableitung von Soll-Modellen aus Ist-Modellen, werden zwar erwähnt, aber nicht näher erläutert. Analysen werden rudimentär durch vordefinierte Abfragen im Modellierungswerkzeug beschrieben und zusätzlich durch eine Werkzeugfunktion für den paarweisen Vergleich von Modellen unterstützt.

Anforderung 13 (Status der Modelle)	Bewertung: +
--	--------------

In Kooperationsbildern und IT-Landschaften können Modell-Status und eine Änderungshistorie explizit als Modellattribute dokumentiert werden.

Tabelle B.5: Bewertung eGPM

Leistungsfall bearbeiten Ist	Ist-Modell B ?
Letzter Änderungsvermerk: 19.09.2014 S. Hofer Initiale Version	
Dokumente zu einem Leistungsfall sind in der Poststelle eingegangen. Die Digitalisierung läuft problemlos und der Fall kann einer versicherten Person zugeordnet werden.	

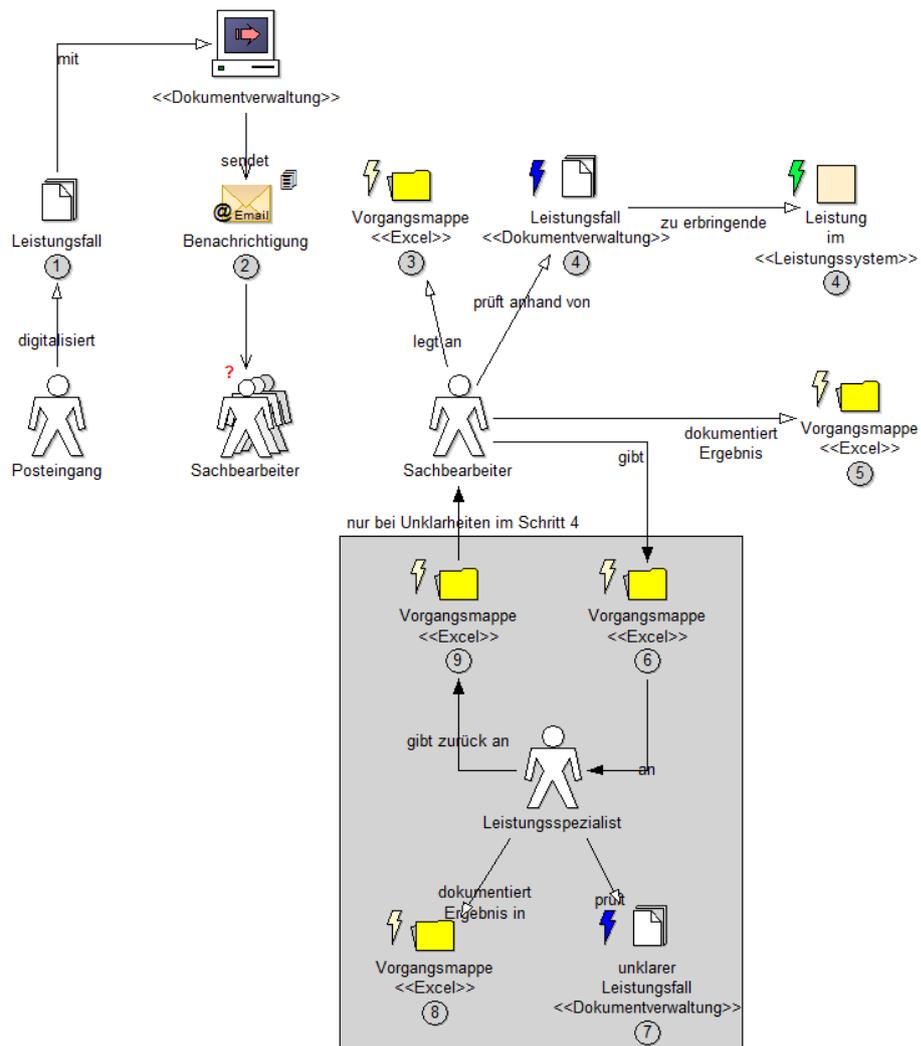


Abbildung B.23: Geschäftsprozess „Leistungsfall bearbeiten“ (Ist) als eGPM Kooperationsbild

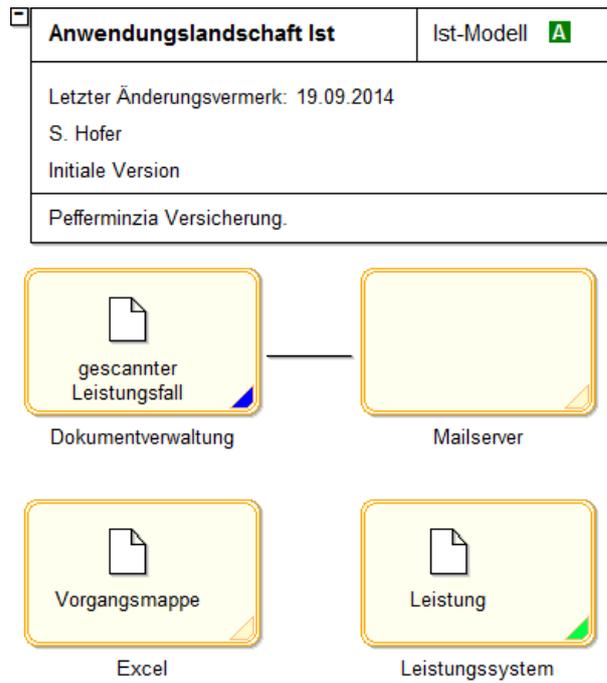


Abbildung B.24: Fachliche Architektur (Ist) als eGPM IT-Landschaft

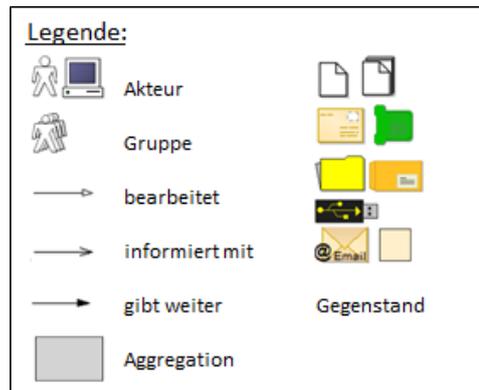


Abbildung B.25: Legende zum Modelltyp Kooperationsbild

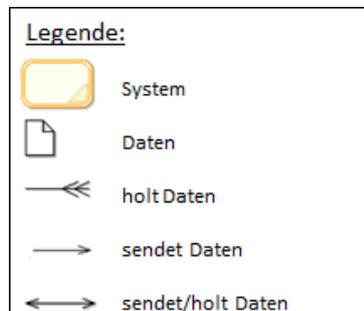


Abbildung B.26: Legende zum Modelltyp IT-Landschaft

Leistungsfall bearbeiten Soll	Soll-Modell B
Letzter Änderungsvermerk: 20.09.2014 S. Hofer Soll-Prozess auf Basis des Ist-Prozesses erstellt	
Dokumente zu einem Leistungsfall sind in der Poststelle eingegangen. Die Digitalisierung läuft problemlos und der Fall kann einer versicherten Person zugeordnet werden.	

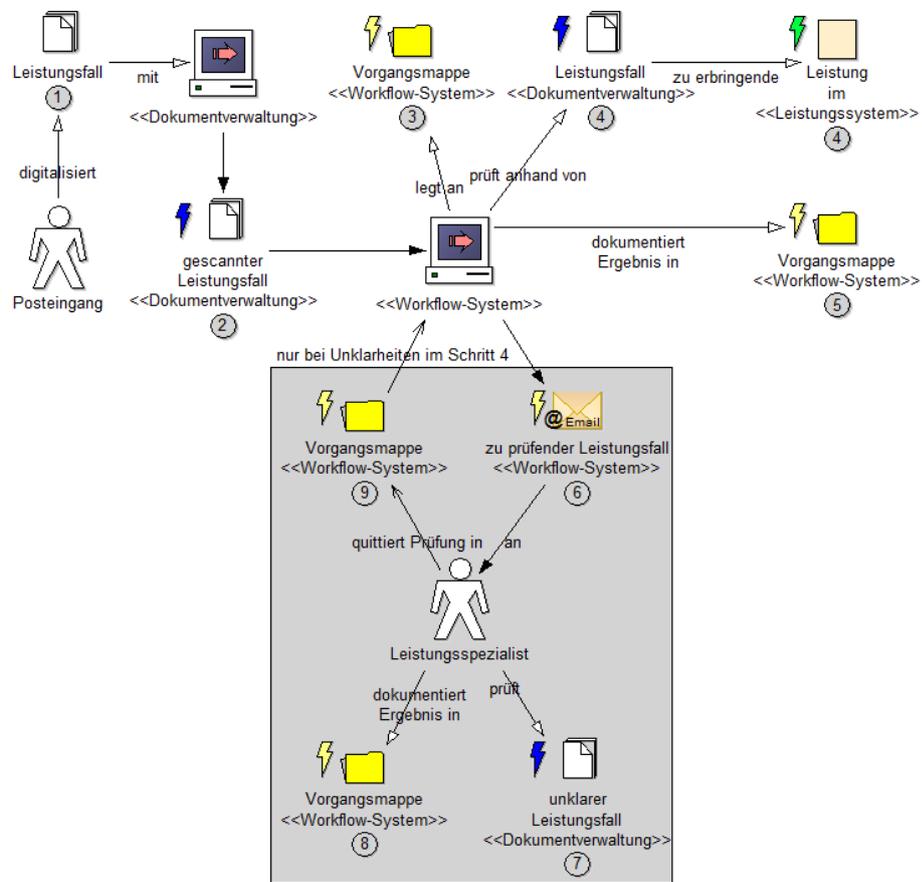


Abbildung B.27: Geschäftsprozess „Leistungsfall bearbeiten“ (soll) als eGPM Kooperationsbild

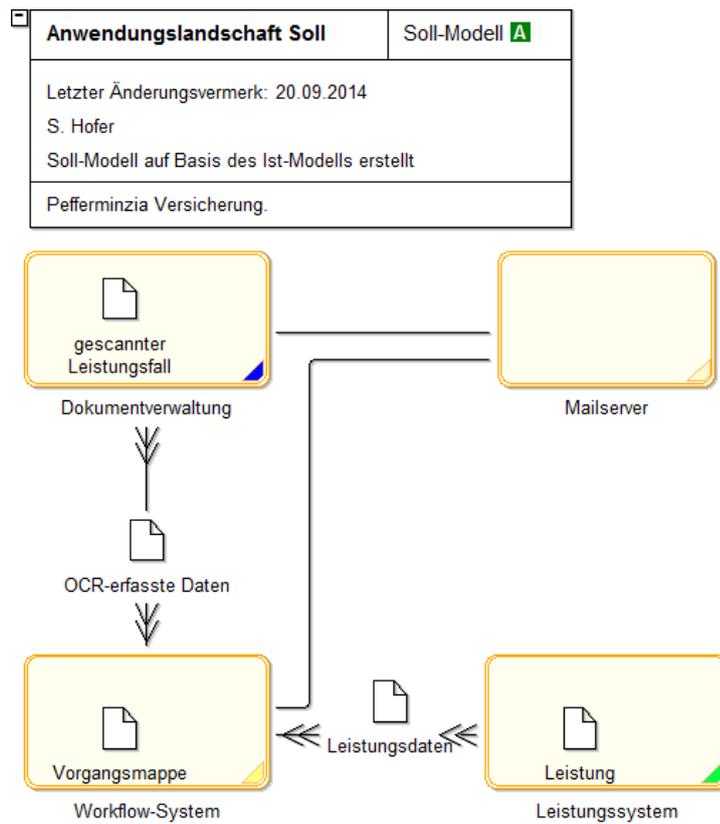


Abbildung B.28: Fachliche Architektur (Soll) als eGPM IT-Landschaft



Abbildung B.29: Begriffsmodell (Ist und Soll) als eGPM Begriffsmodell

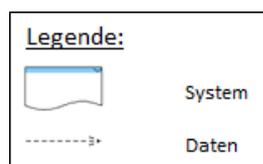


Abbildung B.30: Legende zum Modelltyp Begriffsmodell

Anhang C

eGPM-Modelltyp Kooperationsbild

Die folgende Beschreibung stützt sich zum Teil auf Veröffentlichungen, die unter Mitarbeit des Autors entstanden sind: Eine Kurzbeschreibung der eGPM für ein Fachmagazin (siehe [BH12]) und das Benutzerhandbuch (siehe [WPS14]). Die abgebildeten Beispiele wurden vom Autor erstellt und zeigen Aspekte der fiktiven *Pfefferminzia Versicherung*.

Kooperationsbilder dienen zur Darstellung kooperativer manueller und IT-gestützter Geschäftsprozesse. Sie unterscheiden sich von Prozessmodellierungssprachen, die eine Algorithmus-ähnliche Beschreibung aller möglichen Prozessverläufe in einem Modell ermöglichen. Im Gegensatz dazu stellt ein Modell vom Typ „Kooperationsbild“ jeweils nur ein Szenario dar – das heißt, nur einen der möglichen Verläufe, die ein Prozess nehmen kann. Um eine Szenario-basierte Darstellung zu erreichen, wird auf Fallunterscheidungen und Schleifen verzichtet. Der Modellierer legt stattdessen konkrete Szenarien fest, um den zu modellierenden Prozess einzuschränken. Die Reihenfolge der einzelnen Schritte im Prozessverlauf wird durch Nummerierung festgelegt, und nicht wie in anderen Modellierungssprachen durch die grafische Anordnung der Modellelemente (z.B. von oben nach unten).

Im Mittelpunkt der Kooperationsbilder stehen Menschen mit ihren Aufgaben und ihren Arbeitsgegenständen – nicht Datenflüsse und Funktionen. Kooperationsbilder können z. B. folgende Fragen beantworten:

- Wer sind die handelnden Personen und Softwaresysteme?
- Was tun sie typischerweise in welcher Reihenfolge?
- Mit wem arbeiten sie zusammen?

- Mit welchen Gegenständen oder Informationen arbeiten sie?
- Wann geben sie Gegenstände oder Informationen an wen weiter?
- Bei welchen Tätigkeiten werden sie wie von Informationssystemen unterstützt?

Jeder, an einem Szenario beteiligte Akteur wird nur einmal im Kooperationsbild dargestellt. Modellelemente anderen Typs wie zum Beispiel Gegenstände können mehrfach in einem Kooperationsbild vorkommen, selbst wenn es sich dabei immer um denselben realen Gegenstand handelt. Modellelemente können untereinander mit typisierten Pfeilen verbunden werden, um Tätigkeiten darzustellen. Bei einigen Modellelementtypen kann zwischen verschiedenen Darstellungen für ihre Exemplare gewählt werden, was dem Modellierer einen großen Gestaltungsspielraum lässt. Tabelle C.1 beschreibt die in Kooperationsbildern verwendbaren Modellelementtypen.

Abbildung C.1 zeigt ein Beispiel für ein Kooperationsbild. Gegenstand des Modells ist der Geschäftsprozess „Antragserfassung“ einer Versicherung. Das Kooperationsbild zeigt ein Szenario, in dem die Antragsdaten elektronisch (statt auf Papier) übertragen werden und alle benötigten Informationen vorhanden und korrekt sind (im Gegensatz zu unvollständig oder fehlerhaft ausgefüllten Anträgen). Als Prosatext ausgedrückt lauten die ersten drei Schritte des Prozesses folgendermaßen:

1. Ein Makler schickt mittels des Angebotsrechners einen elektronischen Antrag (mit Antragskennzeichen) an den Host der Versicherung.
2. Der Host erzeugt daraufhin mehrere Datensätze (Antragsdaten, Antragskennzeichen und Provisionsdaten) für den eingegangenen Antrag.
3. Der Makler scannt die zum Antrag gehörigen, begleitenden Dokumente. Diese tragen einen Aufkleber mit dem Antragskennzeichen. Anhand des Antragskennzeichens können der Antrag und die Dokumente einander zugeordnet werden.

Dieses Beispiel zeigt zwei Schwächen der grafischen Darstellung von Kooperationsbildern auf:

- Manche Informationen können in Kooperationsbildern nur schlecht explizit dargestellt werden. Beispielsweise ist nur implizit ersichtlich, dass die vom Host erzeugten Datensätze zum eingegangenen elektronischen Antrag gehören.

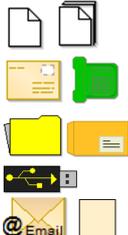
Notation	Name	Beschreibung
	Akteur	Handelnde Personen und Softwaresysteme.
	Gegenstand	Arbeitsgegenstände und -ergebnisse (wie Daten, Dokumente und Medien), Informationen <i>über</i> Arbeitsgegenstände und Informationen zur Koordination von Tätigkeiten.
	Geschäftsfall	Über einen Geschäftsfall kann eine Verknüpfung zu einem anderen Kooperationsbild hergestellt werden. So lässt sich z.B. auf Folgeprozesse verweisen.
	Aggregation	Dient zum Zusammenfassen beliebiger Elemente und kann vielseitig verwendet werden.
	Notiz	Für Randbemerkungen, Kommentare etc.
	informiert mit Gegenstand	Ein Akteur informiert einen anderen Akteur über ein Medium.
	gibt Gegenstand weiter	Ein Gegenstand wird von einem Akteur zu einem anderen weitergereicht.
	bearbeitet	Ein Akteur bearbeitet einen Gegenstand.
	initiiert	Eine flexibel einsetzbare Beziehung zwischen beliebigen Elementen, der durch Beschriftung eine spezielle Bedeutung gegeben werden kann.

Tabelle C.1: Modellelementtypen des Kooperationsbildes

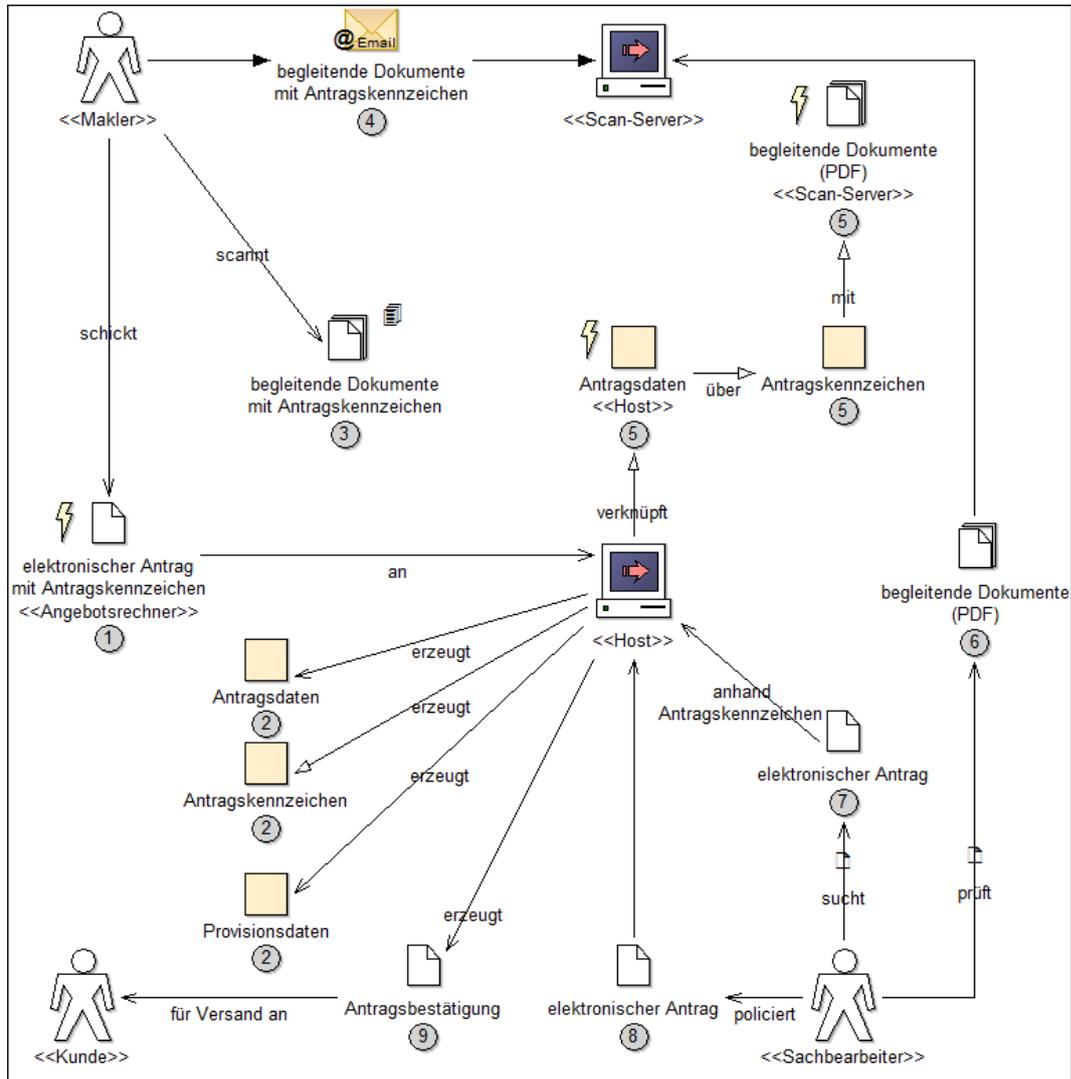


Abbildung C.1: Beispiel für ein Kooperationsbild

Im Prosatext zu Schritt 2 tritt dieser Zusammenhang deutlich hervor („... für den eingegangenen Antrag“).

- An einigen der Schritte (z.B. Schritt 3) zeigen kleine Symbole an, dass im Modellierungswerkzeug textuelle Information hinterlegt wurde, die nur mit Hilfe des Modellierungswerkzeugs angezeigt werden kann. Wird das Modell als Grafik (z.B. im JPG-Format) aus dem Modellierungswerkzeug exportiert, sind die hinterlegten Informationen nicht im Bild enthalten.

Um diese Schwächen zu kompensieren, werden Kooperationsbilder häufig nicht nur grafisch, sondern zusätzlich in Form einer Tabelle beschrieben (vgl. [BH12, S.9]). Diese enthält pro Schritt die Schrittnummer, eine textuelle Beschreibung und eventuell im Werkzeug hinterlegte Kommentare.

Anhang D

Anwendungsbeispiel: Erweiterung von ADOit

Die folgenden Abschnitte beschreiben die vier Modelltypen des vom Autor erweiterten Modellierungsansatzes *ADOit 6.0*. Je Modelltyp fasst eine Abbildung den Zweck und die Syntax des Modelltyps zusammen. Diese Abbildungen sind als Referenz für Modellierer gedacht. Zusätzlich beschreiben textuelle Modellierungsregeln einige syntaktische Besonderheiten, die per Konventionen gelten sollen. Diese Modellierungsregeln konnten nicht über das Metamodell des Modellierungswerkzeugs definiert werden.

Für Werkzeug- und Methodenentwickler werden die Modellierungsregeln mit Hilfe von UML Klassendiagrammen und der *Object Constraint Language (OCL)*, (siehe [Obj14]) formal spezifiziert, sodass eine automatische Regelprüfung implementiert werden kann. Wird eine solche Prüfung implementiert, muss dabei beachtet werden, syntaktisch nicht korrekte Zwischenzustände zu erlauben. Solche Zwischenzustände treten während des werkzeuggestützten Modellierens auf und lassen sich nicht vermeiden.

In den OCL-Ausdrücken werden mehrere Attribute von Modellelementtypen verwendet. Alle Modellelementtypen verfügen über ein textuelles Attribut „name“, das Exemplare eines Modellelementtyps eindeutig identifiziert. Die Modellelementtypen „Arbeitsgegenstand“ und „referenziertes Szenario“ haben ein ganzzahliges Attribut „nummer“. Die übrigen Attribute ergeben sich aus den Relationen, die in den Klassendiagrammen abgebildet sind.

D.1 Kooperationszenario

Abbildung D.1 fasst den Modelltyp zusammen. Folgende Abschnitte beschreiben die Modellierungsregeln mit Hilfe der OCL und eines Klassendiagramms, das zur besseren Lesbarkeit in mehreren Abbildungen aufgeteilt wurde (siehe Abbildungen D.2–D.4). Die mit OCL definierten Constraints beziehen sich auf dieses Klassendiagramm.

Modellierungsregel 1. „kommuniziert über“ ist eine dreistellige Relation und darf nur paarweise verwendet werden.

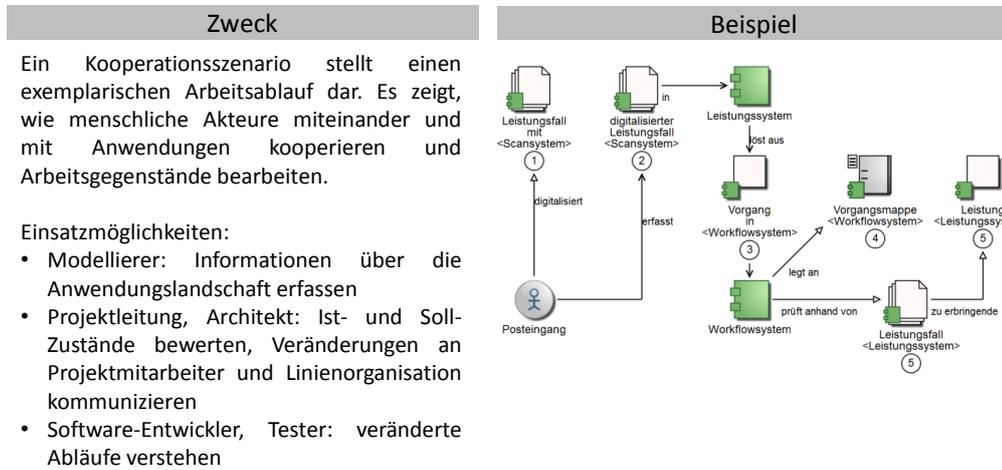
Konzeptionell betrachtet ist eine „kommuniziert über“-Relation dreistellige – ein Akteur oder eine Anwendung kommuniziert über einen Arbeitsgegenstand mit einem anderen Akteur oder einer anderen Anwendung. ADOit unterstützt keine dreistelligen Relationstypen, weswegen „kommuniziert über“ als zweistelliger Relationstyp realisiert wurde. Daher dürfen „kommuniziert über“-Relationen immer nur paarweise modelliert werden (siehe Abbildung D.4). Ein Relations-Paar besteht aus einer Relation *zu* einem Arbeitsgegenstand und einer *von* einem Arbeitsgegenstand wegführenden Relation. Mittels OCL werden zwei Unterarten der Relation definiert (siehe Definitionen *istTeilrelationZuArbeitsgegenstand* und *istTeilrelationVonArbeitsgegenstand*). Die beiden Unterarten schließen sich wechselseitig aus (siehe Invariante *disjunkteTeilrelationen*).

Um kompaktere grafische Modelle zu ermöglichen, kann ein Arbeitsgegenstand in einem Schritt an mehrere Akteure und Anwendungen kommuniziert werden. Dazu muss die Regel der „paarweise Relation“ gelockert werden. Deswegen ist zu jedem Anfang nicht *ein*, sondern *mindestens ein* Ende notwendig (siehe Invariante *zuJedemAnfangMindestensEinEnde*). Ausgangspunkt für eine Relation darf hingegen immer nur einen geben (siehe Invariante *zuJedemEndeEinAnfang*).

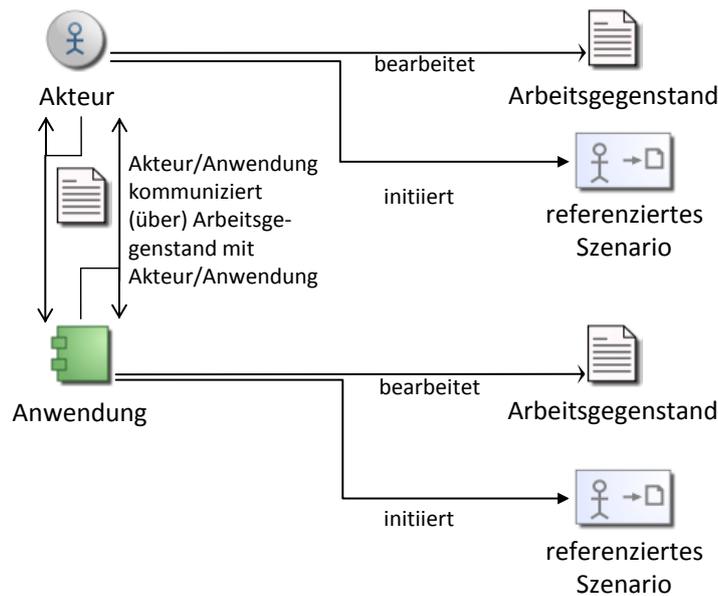
```
context kommuniziert_über
  def: istTeilrelationZuArbeitsgegenstand : Boolean =
    (self.anfang.oclIsTypeOf(Akteur) or
     self.anfang.oclIsTypeOf(Anwendung)) and
     self.teilrelationEnde.oclIsTypeOf(Arbeitsgegenstand)

  def: istTeilrelationVonArbeitsgegenstand : Boolean =
```

Kooperationsszenario



Modellelementtypen und grafische Relationen



Nicht-grafische Relationen



Abbildung D.1: Modelltyp Kooperationsszenario

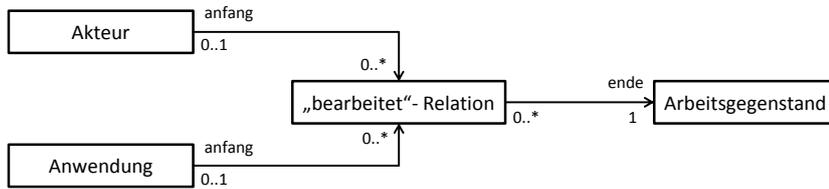


Abbildung D.2: Die Relation „bearbeitet“ als UML Klassendiagramm

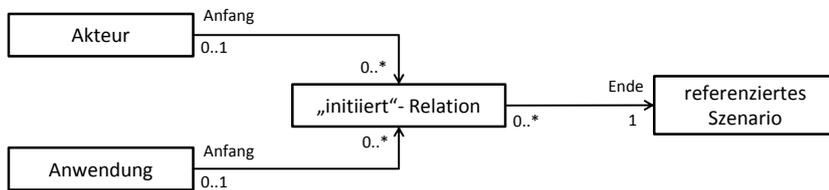
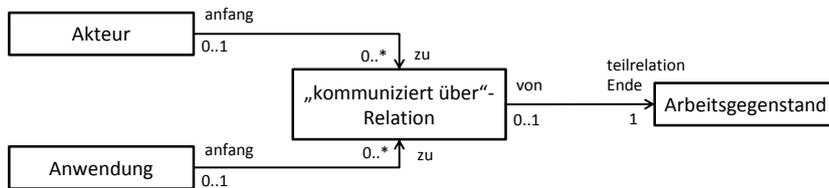
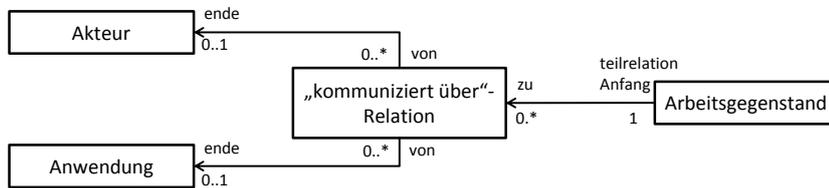


Abbildung D.3: Die Relation „initiiert“ als UML Klassendiagramm



(a) Teilrelation vom Sender zum Arbeitsgegenstand



(b) Teilrelation vom Arbeitsgegenstand zum Empfänger

Abbildung D.4: Die Relation „kommuniziert über“ als UML Klassendiagramm

```

self.teilrelationAnfang.oclIsTypeOf(Arbeitsgegenstand)
and (self.ende.oclIsTypeOf(Akteur) or
self.ende.oclIsTypeOf(Anwendung))

```

```

inv: disjunkteTeilrelationen:
  (istTeilrelationVonArbeitsgegenstand implies
not istTeilrelationZuArbeitsgegenstand)
and
  (istTeilrelationZuArbeitsgegenstand implies
not istTeilrelationVonArbeitsgegenstand)

```

```

inv: zuJedemAnfangMindestensEinEnde:
  istTeilrelationZuArbeitsgegenstand implies
self.teilrelationEnde.zu->
exists(istTeilrelationVonArbeitsgegenstand)

```

```

inv: zuJedemEndeEinAnfang:
  istTeilrelationVonArbeitsgegenstand implies
self.teilrelationAnfang.von->
one(istTeilrelationZuArbeitsgegenstand)

```

Modellierungsregel 2. Die Abfolge von Tätigkeiten in einem Arbeitsablauf wird durch nummerierte Arbeitsgegenstände und referenzierte Szenarien ausgedrückt.

Szenario-basiert zu modellieren heißt, die Tätigkeiten in einer bestimmten Reihenfolge darzustellen. Die Nummerierung entspricht der monoton steigenden Folge der natürlichen Zahlen, beginnend mit 1. Die Folge ist nicht streng monoton steigend (siehe Invariante *arbeitsablaufIstMonotonSteigend*). Die mehrfache Vergabe einer Nummer ist erlaubt und drückt aus, dass die Tätigkeiten gleichzeitig durchgeführt werden.

```

context Arbeitsgegenstand
def: nummernArbeitsgegenstaende : Bag =
  Arbeitsgegenstand.allInstances()->
  collect(a: Arbeitsgegenstand | a.Nummer)

```

```

def: nummernReferenzierteSzenarien : Bag =
    ReferenziertesSzenario.allInstances()->
    collect(r: ReferenziertesSzenario | r.Nummer)

def: arbeitsablauf : OrderedSet =
    nummernArbeitsgegenstaende->
    union(nummernReferenzierteSzenarien)
    .asOrderedSet()

def: vorgaenger : Integer = self.Nummer-1

inv: arbeitsablaufBeginntMit1 : arbeitsablauf->first() = 1

inv: arbeitsablaufIstMonotonSteigend:
    arbeitsablauf->exists(vorgaenger) or (vorgaenger = 0)

```

Modellierungsregel 3. Um die Kooperation zu veranschaulichen, darf jeder Akteur und jede Anwendung nur einmal in einem Modell enthalten sein und jeder Arbeitsgegenstand darf nur in einer einzigen Relation verwendet werden.

```

context Akteur
    inv: akteurIstEindeutig:
        Akteur.allInstances()->isUnique(name)

context Anwendung
    inv: anwendungIstEindeutig:
        Anwendung.allInstances()->isUnique(name)

```

Für Arbeitsgegenstände kann die Regel nicht so einfach formalisiert werden. Denn syntaktisch betrachtet kann eine semantische Relation zu einem Arbeitsgegenstand aus mehreren Pfeilen bestehen:

- **bearbeitet:** Mehrere Akteure bearbeiten gemeinsam einen Arbeitsgegenstand.
- **kommuniziert über:** Über einen Arbeitsgegenstand wird mit mehreren Personen kommuniziert.

Deswegen kann formal nur eingeschränkt geprüft werden, ob ein Arbeitsgegenstand nur in einer Relation verwendet wird: Ein Exemplar eines Arbeitsgegenstands kann entweder in einer bearbeitet oder eine „kommuniziert über“-Relation verwendet werden, aber nicht in beiden:

```
context Arbeitsgegenstand
  inv: disjunkteRelationBearbeitet:
    self.bearbeitet->notEmpty()
    implies (self.von->isEmpty() and self.zu->isEmpty())

  inv: disjunkteRelationKommuniziert:
    (self.von->notEmpty() or self.zu->notEmpty())
    implies self.bearbeitet->isEmpty()
```

Diese Invarianten decken eine weitere Modellierungsregel ab:

Modellierungsregel 4. Jeder Arbeitsgegenstand muss Teil einer Relation sein.

„Verwaiste“ Arbeitsgegenstände sind nicht erlaubt, weil sie in Kooperationsbildern immer im Kontext einer Tätigkeit modelliert werden.

Modellierungsregel 5. Referenzierte Szenarien müssen von einem Akteur oder einer Anwendung über eine „initiiert“-Relation angestoßen werden.

Mit referenzierten Szenarien werden Verknüpfungen zu Teil- oder Folgeprozessen dargestellt. Da jeder Prozess einen Auslöser hat, müssen referenzierte Szenarien von einem Akteur oder einer Anwendung über eine „initiiert“-Relation angestoßen werden:

```
context Referenziertes_Szenario
  inv: hatRelation:
    self.initiiertRelation->notEmpty()
```

D.2 Anwendungsarchitektur

Abbildung D.5 zeigt die Zusammenfassung des Modelltyps. Folgende Modellierungsregel gilt:

Modellierungsregel 6. Jede Anwendungen darf nur einmal in einem Modell enthalten sein.

Anwendungen sind die zentralen Bausteine einer fachlichen Architektur. Um eine solche mit Hilfe des Modelltyps Anwendungsarchitektur darzustellen, ist es sinnvoll, wenn jede Anwendung nur einmal pro Modell enthalten ist. Modellelemente anderer Typen (wie zum Beispiel Schnittstellen und Services) können hingegen mehrfach vorkommen, um die grafische Übersichtlichkeit zu erhöhen.

context Anwendung

inv: anwendungIstEindeutig:

Anwendung.allInstances()->isUnique(name)

D.3 Datenarchitektur

Abbildung D.6 zeigt die Zusammenfassung des Modelltyps. Folgende Modellierungsregel gilt:

Modellierungsregel 7. Jedes Geschäftsobjekt darf nur einmal in einem Modell enthalten sein.

Damit ein Modell vom Typ Datenarchitektur als Glossar verwendet werden kann, müssen die darin modellierten Konzepte eindeutig sein.

context Geschaeftsobjekt

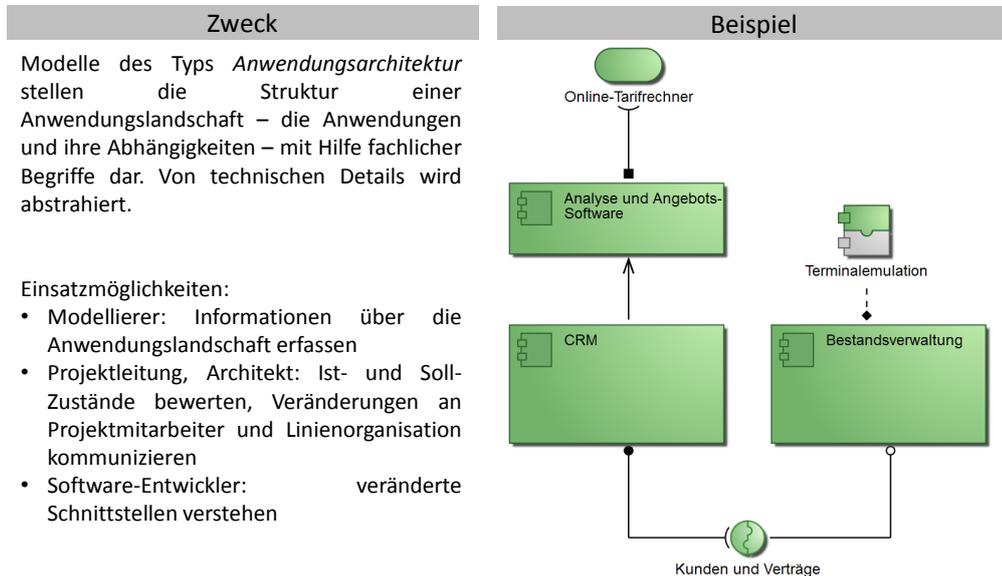
inv: geschaeftsobjektIstEindeutig:

Geschaeftsobjekt.allInstances()->isUnique(name)

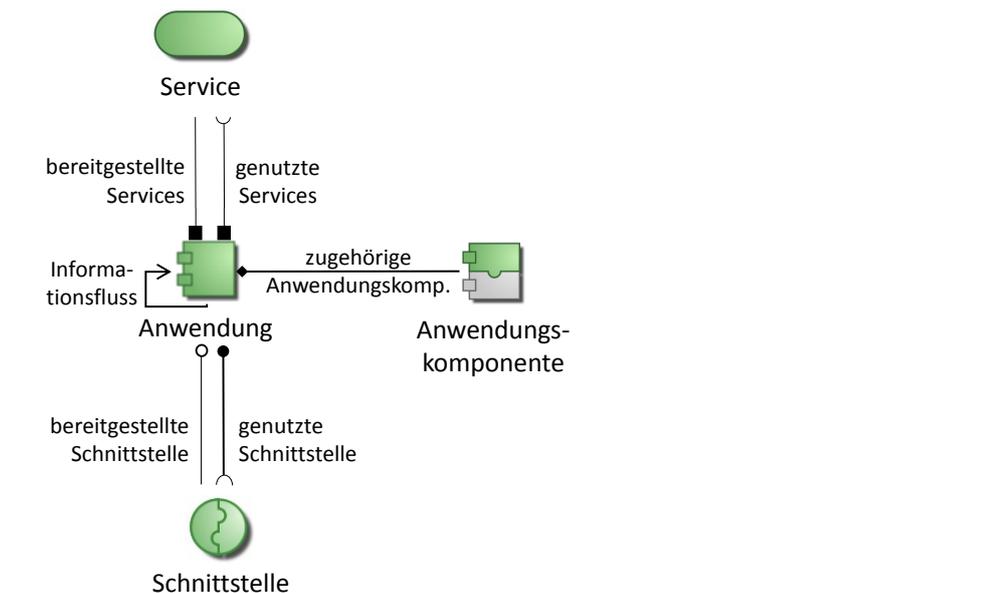
D.4 Technologiearchitektur

Abbildung D.7 zeigt die Zusammenfassung des Modelltyps. Dieser Modelltyp muss flexibel genug sein, um beispielsweise verteilte, virtuelle und redundante Infrastruktur darstellen zu können. Um die Flexibilität zu wahren, werden keine Modellierungsregeln für Technologiearchitekturen spezifiziert.

Anwendungsarchitektur



Modellelementtypen und grafische Relationen



Nicht-grafische Relationen

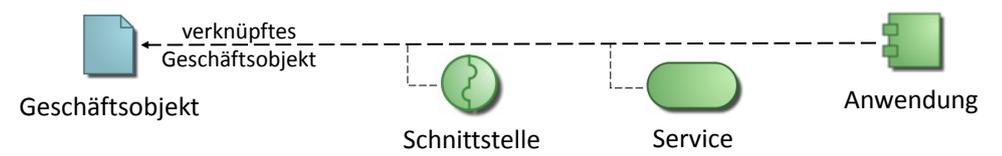
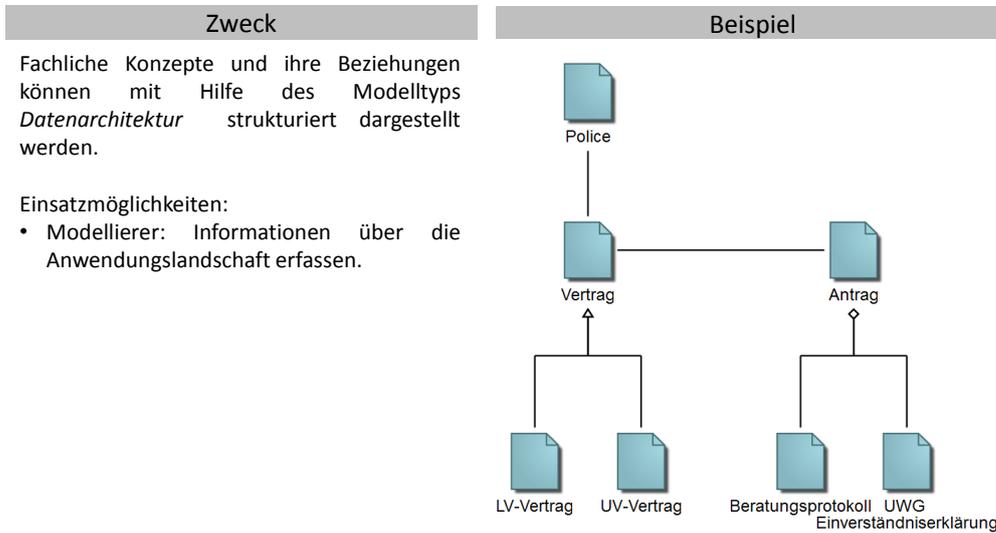


Abbildung D.5: Modelltyp Anwendungsarchitektur

Datenarchitektur



Modellelementtypen und grafische Relationen



Nicht-grafische Relationen

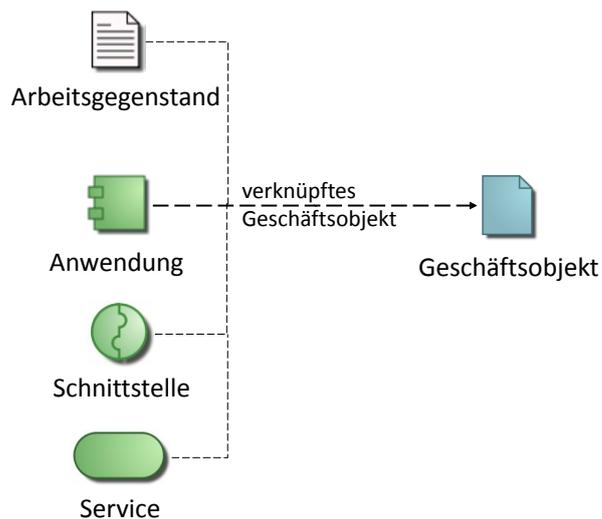


Abbildung D.6: Modelltyp Datenarchitektur

Technologiearchitektur

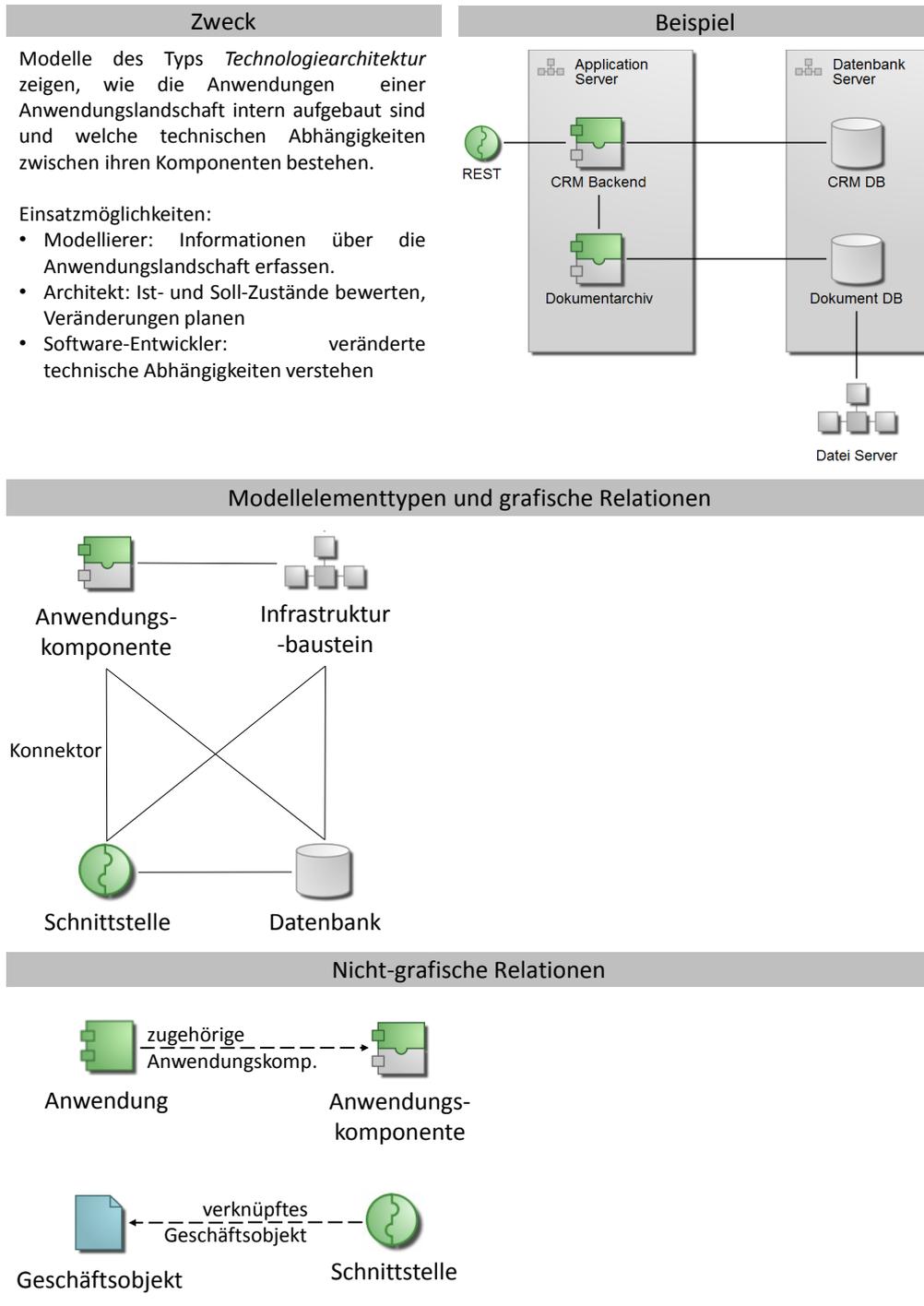


Abbildung D.7: Modelltyp Technologiearchitektur

Anhang E

Evaluation des erweiterten ADOit

Dieser Anhang enthält die Abbildungen, die in Abschnitt 6.2.3 zur Evaluation der Praxistauglichkeit des erweiterten ADOit herangezogen wurden.

- Kooperationszenario „Antragsprozess für eine Krankenversicherung“ (siehe Abbildung E.1)
- Kooperationszenario „Regelprüfung im Antragsprozess“ (siehe Abbildung E.2)
- Kooperationszenario „Prozess für eine Ehegattenermäßigung“ (siehe Abbildung E.3)
- Kooperationszenario „Krankenleistungsprozess“ (siehe Abbildung E.4)
- Datenarchitektur „Konzepte“ (siehe Abbildung E.5)
- Anwendungsarchitektur „Anwendungslandschaft (KV-Anträge und Leistung)“ (siehe Abbildung E.6)
- Technologiearchitektur „Technische Referenzarchitektur mit WfMS“ (siehe Abbildung E.7)

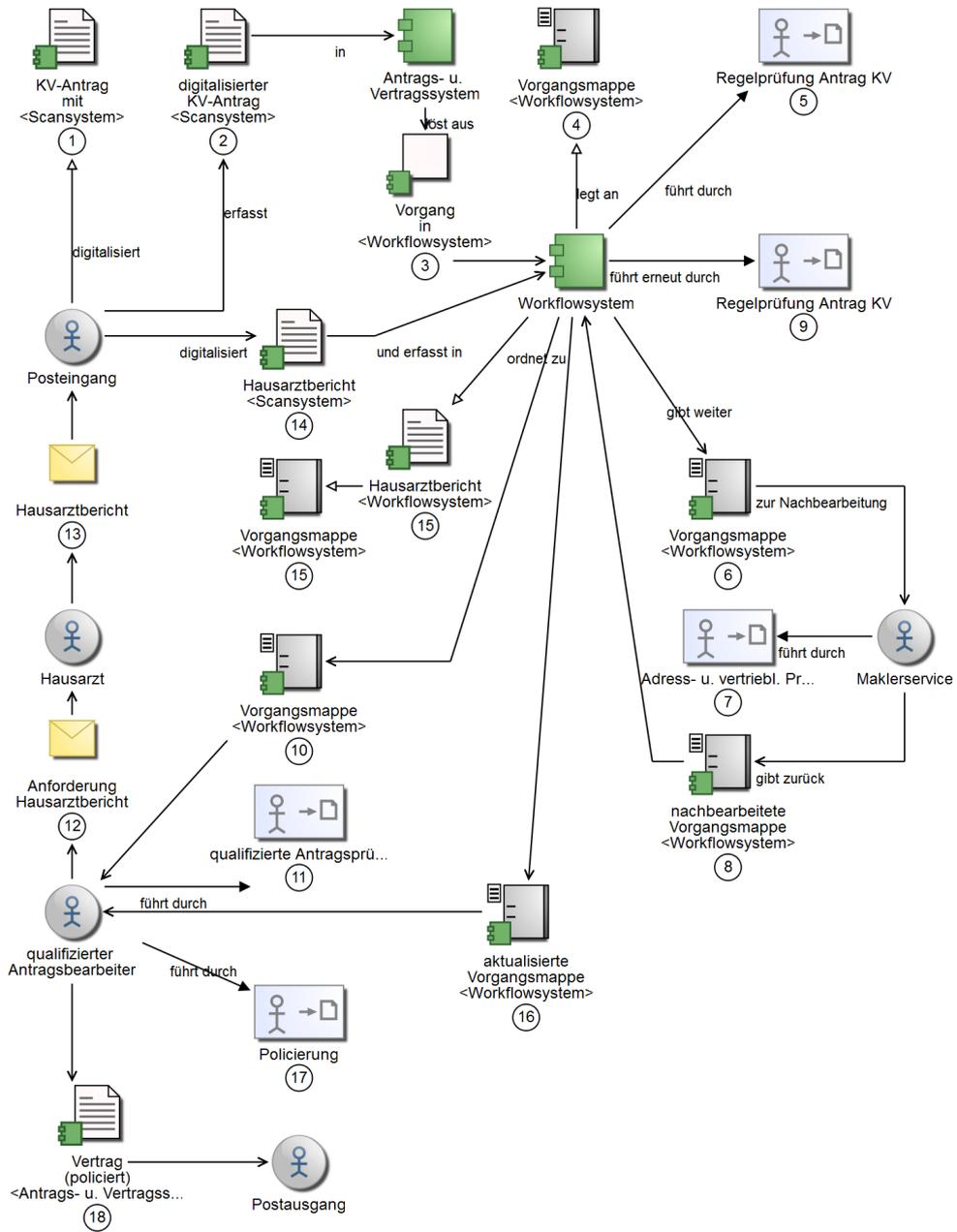


Abbildung E.1: Antragsprozess für eine Krankenversicherung

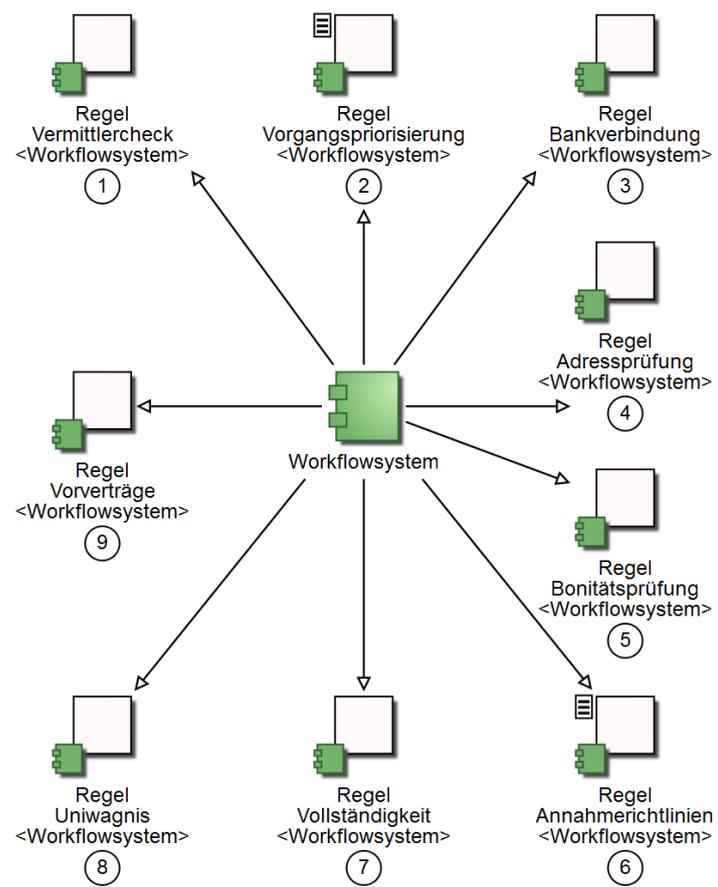


Abbildung E.2: Regelprüfung im Antragsprozess

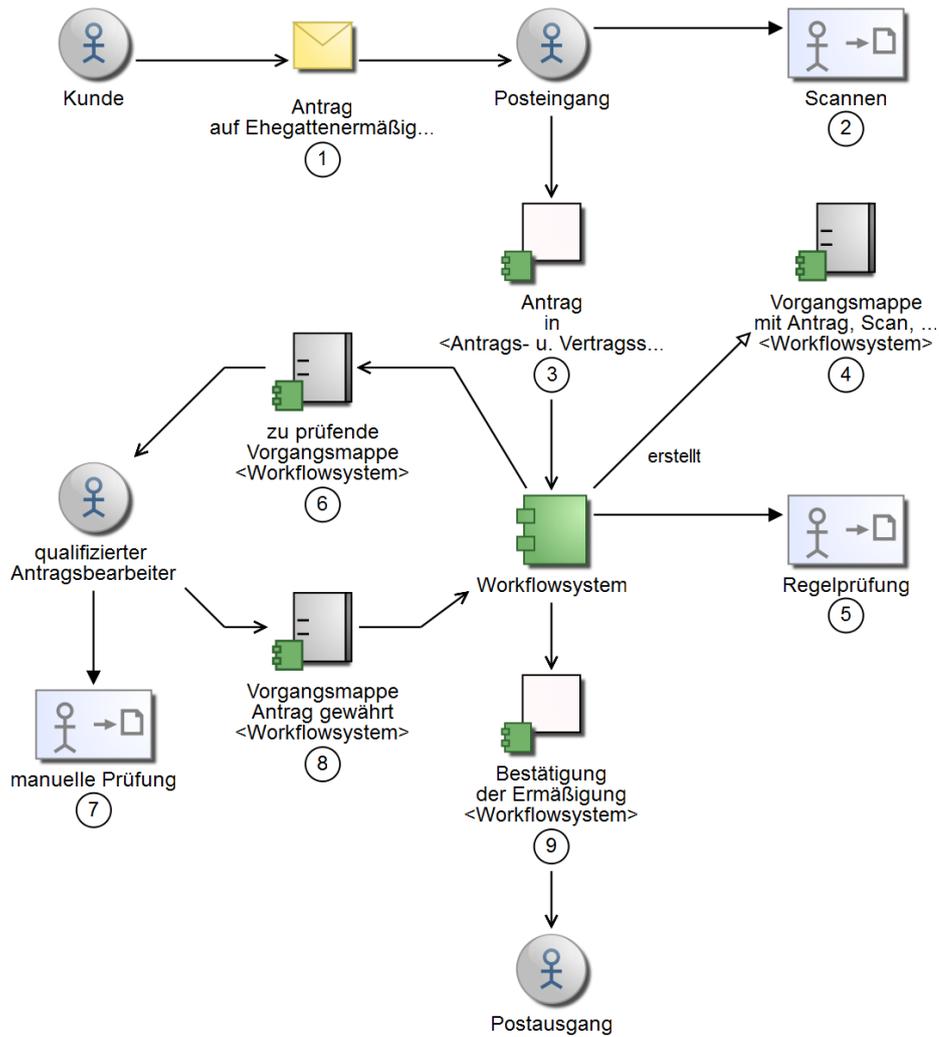


Abbildung E.3: Prozess für eine Ehegattenermäßigung

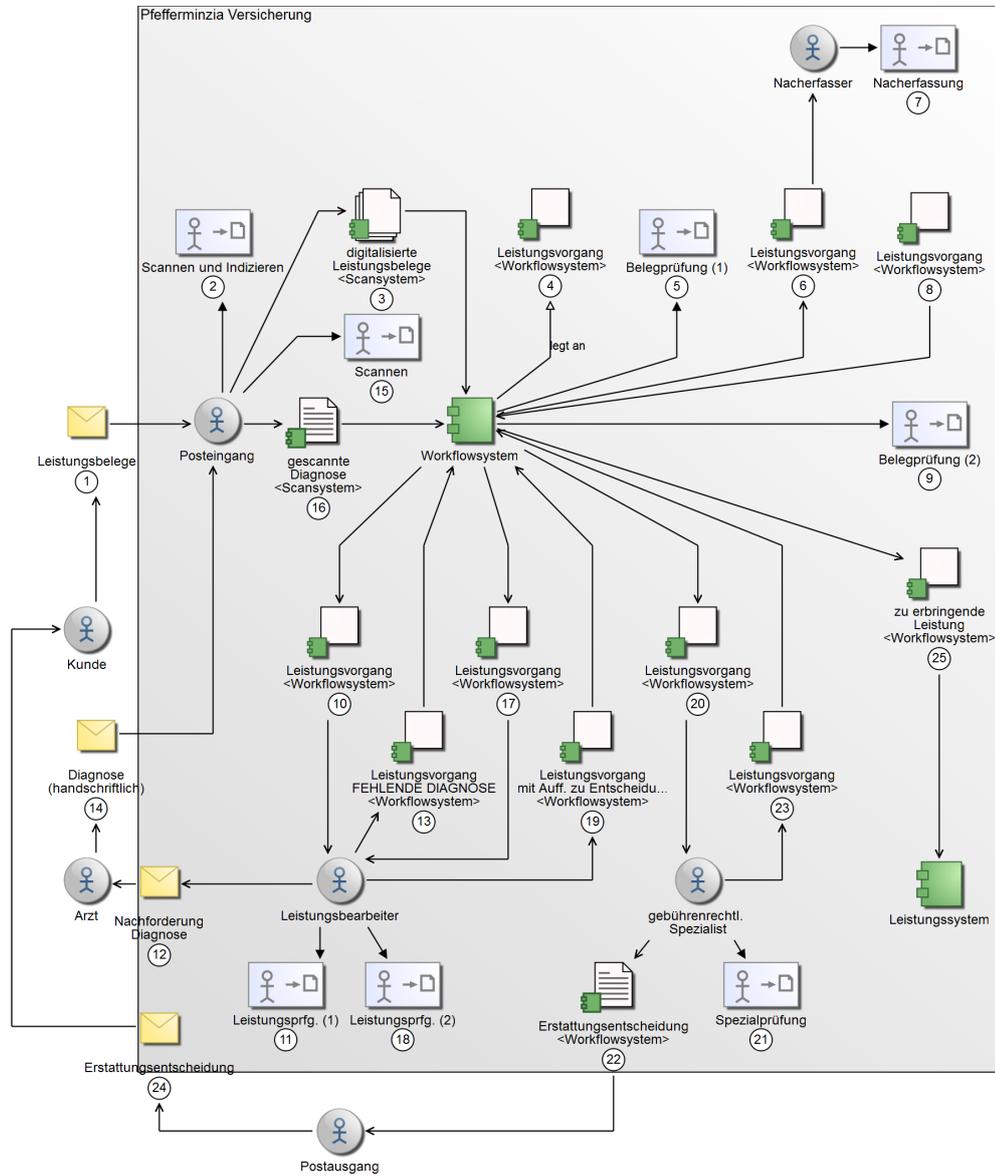


Abbildung E.4: Krankenleistungsprozess

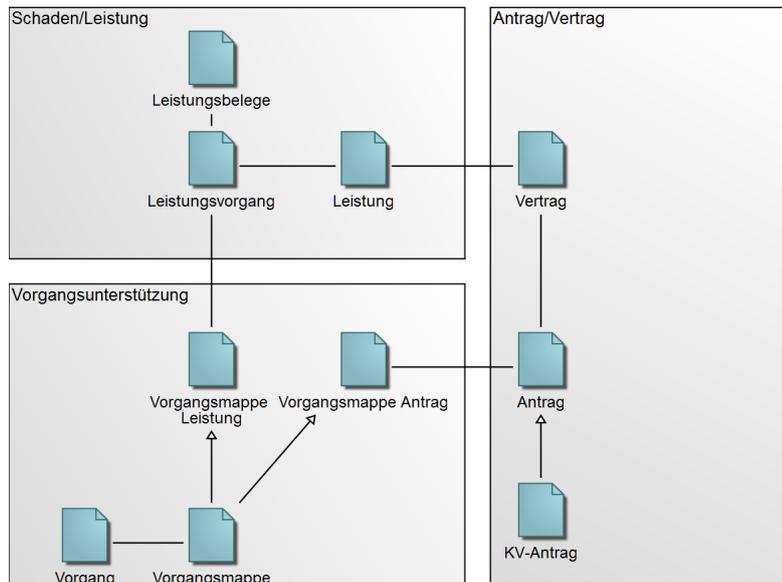


Abbildung E.5: Fachliche Konzepte als Datenarchitektur-Modell

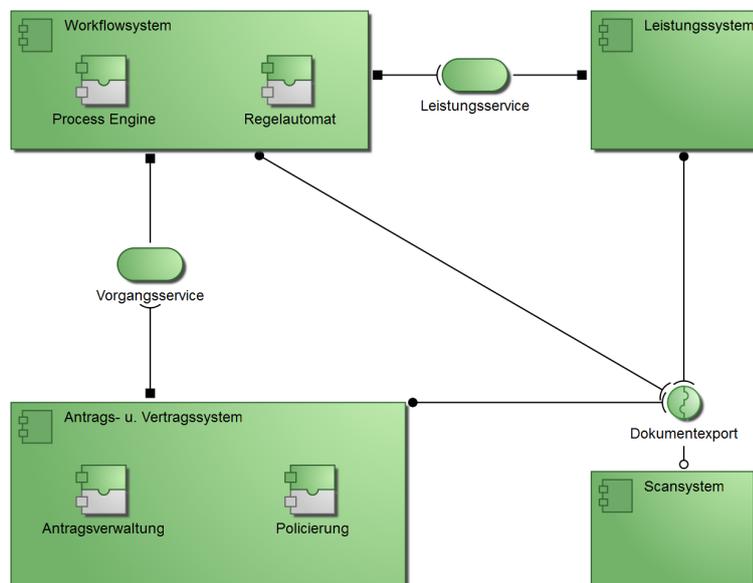


Abbildung E.6: Anwendungslandschaft (KV-Anträge und Leistung) als Anwendungsarchitektur-Modell

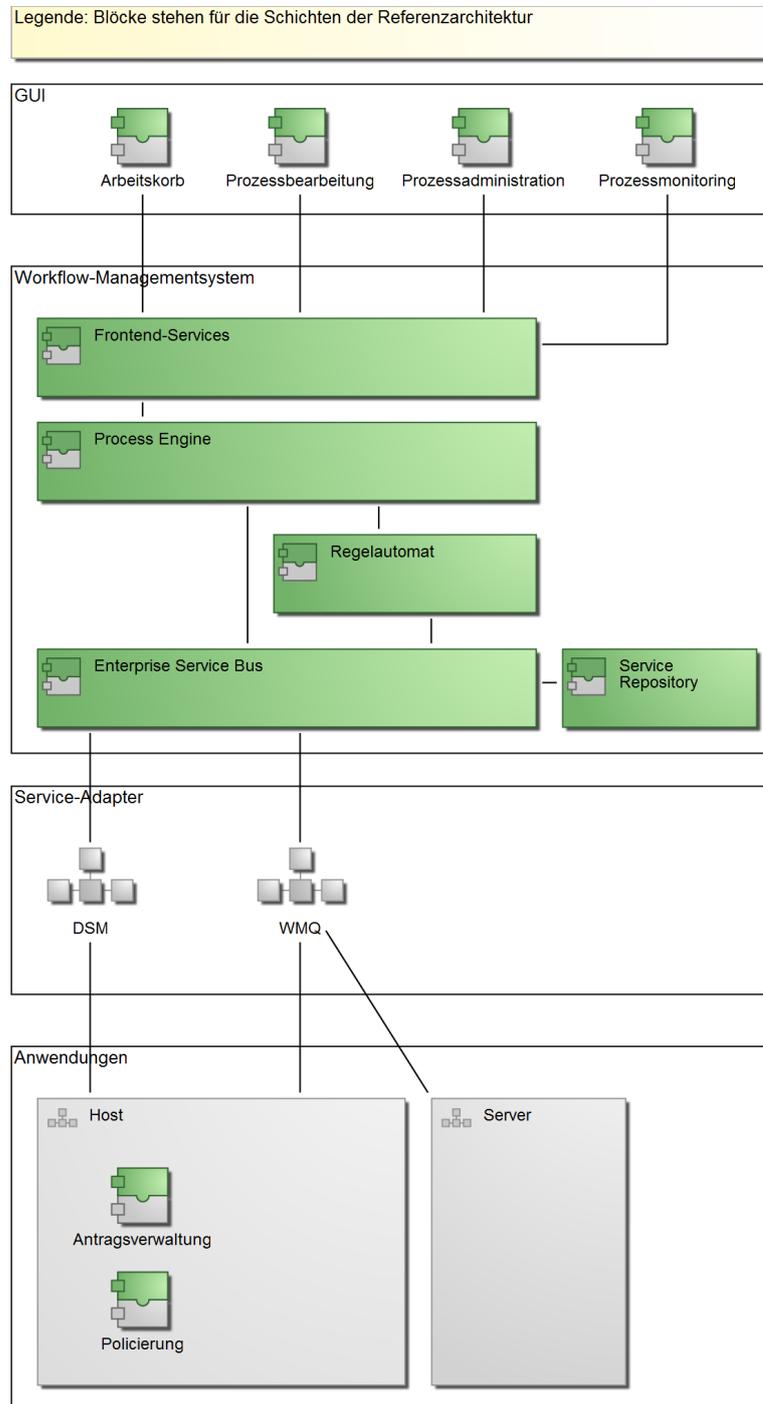


Abbildung E.7: Technische Referenzarchitektur mit WfMS als Technologiearchitektur-Modell

Literaturverzeichnis

- [Aal11] AALST, Will van d.: Intra- and Inter-Organizational Process Mining: Discovering Processes Within and Between Organizations. In: *The Practice of Enterprise Modeling. 4th IFIP WG 8.1 Working Conference, PoEM 2011* Bd. 92, 2011 (LNBIP), S. 1–11
- [ANS00] ANSI/IEEE COMPUTER SOCIETY: *ANSI/IEEE-Standard-1471. IEEE Recommended Practice for Architectural Description of Software-Intensive Systems*, 2000
- [ARW08] AIER, Stephan ; RIEGE, Christian ; WINTER, Robert: Unternehmensarchitektur. Literaturüberblick und Stand der Praxis. In: *Wirtschaftsinformatik* 4 (2008), S. 292–304
- [BCK12] BASS, Len ; CLEMENTS, Paul ; KAZMAN, Rick: *Software Architecture in Practice*. Addison Wesley, 2012
- [BD87] BOX, George ; DRAPER, Norman: *Empirical Model-Building and Response Surfaces*. Wiley, 1987
- [BEL⁺09] BUCKL, Sabine ; ERNST, Alexander M. ; LANKES, Josef ; MATTHES, Florian ; SCHWEDA, Christian M.: State of the Art in Enterprise Architecture Management / Technische Universität München, Lehrstuhl Software Engineering betrieblicher Informationssysteme. 2009. – Forschungsbericht
- [BELM08] BUCKL, Sabine ; ERNST, Alexander M. ; LANKES, Josef ; MATTHES, Florian: Enterprise Architecture Management Pattern Catalog / Lehrstuhl Software Engineering betrieblicher Informationssysteme, Technische Universität München. 2008. – Forschungsbericht

- [BEMS09] BUCKL, Sabine ; ERNST, Alexander ; MATTHES, Florian ; SCHWEDA, Christian: An Information Model for Managed Application Landscape Evolution. In: *Journal of Enterprise Architecture* 5 (2009), Nr. 1, S. 12–26
- [BH12] BREITLING, Holger ; HOFER, Stefan: Beispielhaft gut modelliert: Exemplarische Geschäftsprozessmodellierung in der Praxis. In: *ObjektSpektrum* 6 (2012), S. 8–13
- [Biz] BIZZDESIGN: *Webseite*, <http://www.de.bizzdesign.com/>. – zuletzt besucht am 15.12.2015
- [BKS06] BREITLING, Holger ; KORNSTÄDT, Andreas ; SAUER, Joachim: Design Rationale in Exemplary Business Process Modeling. In: DUTOIT, Alen H. (Hrsg.) ; MCCALL, Ray (Hrsg.) ; MISTRİK, Ivan (Hrsg.) ; PAECH, Barbara (Hrsg.): *Rationale Management in Software Engineering*. Springer, 2006, S. 191–208
- [BOCa] BOC GROUP: *Webseite*, <https://de.boc-group.com/>. – zuletzt besucht am 15.12.2015
- [BOCb] BOC GROUP: *Webseite*, <http://www.adoit-community.com/>. – zuletzt besucht am 15.12.2015
- [BP07] BECKER, Jörg ; PFEIFFER, Daniel: Konzeptionelle Modellierung. In: LEHNER, Franz (Hrsg.) ; ZELEWSKI, Stephen (Hrsg.): *Wissenschaftstheoretische Fundierung und wissenschaftliche Orientierung der Wirtschaftsinformatik*. Gito, 2007, S. 1–17
- [Bra73] BRATEN, Stein: Model Monopoly and Communications: Systems Theoretical Notes on Democratization. In: *Acta Sociologica* 16 (1973), Nr. 2, S. 98–107
- [Bra07] BRAUN, Christian: *Modellierung der Unternehmensarchitektur*, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften, Universität St. Gallen, Diss., 2007
- [Car00] CARROLL, John: *Making Use. Scenario-Based Design of Human-Computer Interaction*. MIT Press, 2000

- [Che76] CHEN, Peter: The Entity-Relationship Model: Toward a Unified View of Data. In: *ACM Transactions on Database Systems* 1 (1976), Nr. 1, S. 9–36
- [Coc00] COCKBURN, Alistair: *Writing Effective Use Cases*. Addison-Wesley Longman, 2000. – ISBN 9780201702255
- [Der03] DERN, Gernot: *Management von IT-Architekturen*. Vieweg, 2003
- [EHH⁺08] ENGELS, Gregor ; HESS, Andreas ; HUMM, Bernhard ; JUNG, Oliver ; LOHMANN, Marc ; RICHTER, Jan-Peter ; VOSS, Markus ; WILLKOMM, Johannes: *Quasar Enterprise*. dpunkt.verlag, 2008
- [FGKW07] FILL, Hans-Georg ; GERICKE, Anke ; KARAGIANNIS, Dimitris ; WINTER, Robert: Modellierung für Integrated Enterprise Balancing. In: *Wirtschaftsinformatik* 49 (2007), Nr. 6, S. 419–429
- [FK98] FLOYD, Christiane ; KLISCHEWSKI, Ralf: Modellierung – ein Handgriff zur Wirklichkeit. Zur sozialen Konstruktion und Wirksamkeit von Informatik-Modellen. In: *Proceedings der Modellierung '98*, 1998, S. 21–26
- [FKKL09] FEIRI, Tim ; KELLER, Stephanie ; KERSCHER, Ingrid ; LEHMANN, Frank: ARIS-Modelltypen. Übersichtslandkarten und Einzelbeschreibungen / Berufsakademie Ravensburg. 2009. – Arbeitsberichte zur Wirtschaftsinformatik
- [For13] FORRESTER RESEARCH: EA Management Suites. In: *The Forrester Wave* (2013), Nr. Q2
- [Fow00] FOWLER, Martin: *Refactoring: Improving the Design of Existing Code*. Addison-Wesley Longman, 2000
- [Fra12] FRANK, Ulrich: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. In: *Journal of Software and Systems Modeling* (2012)
- [FS93] FERSTL, Otto K. ; SINZ, Elmar J.: Der Modellierungsansatz des semantischen Objektmodells (SOM). In: *Bamberger Beiträge zur Wirtschaftsinformatik* (1993), Nr. 18

- [FZ02] FLOYD, Christiane ; ZÜLLIGHOVEN, Heinz: Softwaretechnik. In: RECHENBERG, Peter (Hrsg.) ; POMBERGER, Gustav (Hrsg.): *Informatik-Handbuch*. Hanser, 2002, S. 763–790
- [GKRS12] GHARBI, Mahbouba ; KOSCHE, Arne ; RAUSCH, Andreas ; STARKE, Gernot: *Basiswissen für Software-Architekten*. dpunkt.verlag, 2012
- [Han13] HANSCHKE, Inge: *Strategisches Management der IT-Landschaft*. Hanser, 2013
- [Hei02] HEINRICH, Lutz J.: Grundlagen der Wirtschaftsinformatik. In: RECHENBERG, Peter (Hrsg.) ; POMBERGER, Gustav (Hrsg.): *Informatik-Handbuch*. Hanser, 2002, S. 1039–1054
- [HHL99] HERRMANN, Thomas ; HOFFMANN, Marcel ; LOSER, Kai-Uwe: Modellieren mit SeeMe - Alternativen wider die Trockenlegung feuchter Informationslandschaften. In: *Proceedings der Modellierung '99*, 1999, S. 59–74
- [HM08] HESSE, Wolfgang ; MAYR, Heinrich: Modellierung in der Softwaretechnik: Eine Bestandsaufnahme. In: *Informatik-Spektrum* 31 (2008), Nr. 5, S. 377–393
- [HMPR04] HEVNER, Alan R. ; MARCH, Salvatore T. ; PARK, Jinsoo ; RAM, Sudha: Design Science in Information Systems Research. In: *MIS Quarterly* 28 (2004), S. 75–105
- [Hof09] HOFER, Stefan: Anwendungslandschaften und ihre Verwendung durch exemplarische Geschäftsprozessmodellierung verstehen. In: *Proceedings of the 1. Workshop des GI-Arbeitskreises Langlebige Softwaresysteme (L2S2)* Bd. 537, 2009 (CEUR Workshop Proceedings), S. 27–38
- [Hof11] HOFER, Stefan: Instances Over Algorithms: A Different Approach to Business Process Modeling. In: *The Practice of Enterprise Modeling. 4th IFIP WG 8.1 Working Conference, PoEM 2011* Bd. 92, 2011 (LNBIP), S. 25–37
- [HW04] HOHPE, Gregor ; WOOLF, Bobby: *Enterprise Integation Patterns*. Addison-Wesley, 2004

- [Int] INTERNATIONAL SOFTWARE ARCHITECTURE QUALIFICATION BOARD: *Webseite*, <http://www.isaqb.org/>. – zuletzt besucht am 15.12.2015
- [ISO14] ISO/IEC: *International Standard ISO/IEC 25000:2014 – Software Engineering – Software Product Quality Requirements and Evaluation (SQUARE)*, 2014
- [JEJ95] JACOBSON, Ivar ; ERICSSON, Maria ; JACOBSON, Agneta: *The Object Advantage: Business Process Reengineering with Object Technology*. Addison-Wesley, 1995
- [Kel07] KELLER, Wolfgang: *IT-Unternehmensarchitektur. Von der Geschäftsstrategie zu optimalen IT-Unterstützung*. 1. Auflage. dpunkt.verlag, 2007
- [Kel12] KELLER, Wolfgang: *IT-Unternehmensarchitektur. Von der Geschäftsstrategie zu optimalen IT-Unterstützung*. 2. Auflage. dpunkt.verlag, 2012
- [KK02] KARAGIANNIS, Dimitris ; KÜHN, Harald: Metamodelling Platforms. In: *Proceedings of the Third International Conference EC-Web*, Springer, 2002 (LNCS 2455), S. 182–195
- [KNS92] KELLER, Gerhard ; NÜTTGENS, Markus ; SCHEER, August-Wilhelm: Semantische Prozeßmodellierung auf der Grundlage Ereignisgesteuerter Prozeßketten (EPK). In: *Forschungsberichte des Instituts für Wirtschaftsinformatik, Universität des Saarlandes* (1992), Nr. 89
- [Kru95] KRUCHTEN, Phillipe: The 4+1 View Model of Architecture. In: *IEEE Software* 12 (1995), Nr. 6, S. 42–50
- [KWR96] KRABBEL, Anita ; WETZEL, Ingrid ; RATUSKI, Sabine: Objektorientierte Analysetechniken für übergreifende Aufgaben. In: *GI-Fachtagung Softwaretechnik '96*, 1996, S. 65–72
- [Lan13] LANKHORST, Marc: *Enterprise Architecture at Work. Modelling, Communication. and Analysis*. Springer, 2013
- [Leh80] LEHMAN, Meir: Programs, Life Cycles, and Laws of Software Evolution. In: *Proceedings of the IEEE* 68 (1980), Nr. 9, S. 1060–1076

- [Leh08] LEHMANN, Frank: *Integrierte Prozessmodellierung mit ARIS*. dpunkt.verlag, 2008
- [LG06] LEIST-GALANOS, Susanne: *Methoden zur Unternehmensmodellierung – Vergleich, Anwendungen und Integrationspotenziale*. Logos, 2006
- [Lil08] LILIENHAL, Carola: *Komplexität von Softwarearchitekturen. Stile und Strategien*, Fachbereich Informatik, Universität Hamburg, Diss., 2008
- [LL13] LUDEWIG, Jochen ; LICHTER, Horst: *Software Engineering. Grundlagen, Menschen, Prozesse, Techniken*. dpunkt.verlag, 2013
- [LSS94] LINDLAND, Odd I. ; SINDRE, Guttorm ; SOLVBERG, Arne: Understanding Quality In Conceptual Modeling. In: *IEEE Software* 11 (1994), Nr. 2, S. 42–49
- [LSV00] LAARTZ, Jürgen ; SONDEREGGER, Ernst ; VINCKIER, Johan: The Paris Guide to IT Architecture. In: *The McKinsey Quarterly* (2000), Nr. 3, S. 118–127
- [LZ14] LILIENHAL, Carola ; ZÜLLIGHOVEN, Heinz: Der Turmbau zu Babel: Die Grenzen der Informatik. In: *ObjektSpektrum* (2014), Nr. 1
- [MBDS11] MYKHASHCHUK, Mariana ; BUCKL, Sabine ; DIERL, Thomas ; SCHWEDA, Christian: Charting the Landscape of Enterprise Architecture Management – An Extensive Literature Analysis. In: *Wirtschaftsinformatik Proceedings*, 2011
- [MDN08] MOHAGHEGHI, Parastoo ; DEHLEN, Vegard ; NEPLE, Tor: Towards a Tool-Supported Quality Model for Model-Driven Engineering. In: *Proceedings of the 3rd Workshop on Quality in Modeling* Bd. 5421, 2008 (LNCS), S. 74–88
- [MKP00] MITLETON-KELLY, Eve ; PAPAETHIMIOU, Maria-Christiana: Co-Evolution Of Diverse Elements Interacting Within A Social Ecosystem. In: *International Workshop on Feedback and Evolution in Software and Business Processes (FEAST 2000)*, 2000

- [MRC07] MENDLING, Jan ; REIJERS, Hajo ; CARDOSO, Jorge: What Makes Process Models Understandable? In: *BPM 2007* Bd. 4714, Springer, 2007 (LNCS), S. 48–63
- [MS95] MARCH, Salvatore T. ; SMITH, Gerald F.: Design and Natural Science Research on Information Technology. In: *Decision Support Systems 15* (1995), S. 251–266
- [MW04] MATTHES, Florian ; WITTENBURG, André: Softwarekarten zur Visualisierung von Anwendungslandschaften und ihren Aspekten – eine Bestandsaufnahme / Lehrstuhl Software Engineering betrieblicher Informationssysteme. 2004. – Forschungsbericht
- [Obj14] OBJECT MANAGEMENT GROUP: *Object Constraint Language*, 2014. <http://www.omg.org/spec/OCL/2.4>. – zuletzt besucht am 15.12.2015
- [Oes95] OESTERLE, Hubert: *Business Engineering. Prozeß- und Systementwicklung (Band 1: Entwurfstechniken)*. Springer, 1995
- [Ope] OPEN MODELS INITIATIVE: *Webseite*, www.openmodels.at. – zuletzt besucht am 15.12.2015
- [RJB05] RUMBAUGH, James ; JACOBSON, Ivar ; BOOCH, Grady: *The Unified Modeling Language Reference Manual*. Addison-Wesley, 2005
- [Rol98] ROLF, Arno: *Grundlagen der Organisations- und Wirtschaftsinformatik*. Springer, 1998
- [Ros09] ROST, Johannes: *Vergleichende Evaluation von Workflow-Management-Systemen*, Fachbereich Informatik, Universität Hamburg, Diplomarbeit, 2009
- [Sch02] SCHEER, August-Wilhelm: *ARIS – Vom Geschäftsprozess zum Anwendungssystem*. Springer, 2002
- [Sch05] SCHWINN, Alexander: *Entwicklung einer Methode zur Gestaltung von Integrationsarchitekturen für Informationssysteme*, Hochschule für Wirtschafts-, Rechts- und Sozialwissenschaften, Universität St. Gallen, Diss., 2005

- [Sch09] SCHÖNHERR, Marten: Towards a Common Terminology in the Discipline of Enterprise Architecture. In: *Service-Oriented Computing – ICSOC 2008 Workshops* Bd. 5472, 2009 (LNCS), S. 400–413
- [SHP09] SSEBUGGWAWO, Denis ; HOPPENBROUWERS, Stijn ; PROPER, Erik: Interactions, Goals and Rules in a Collaborative Modelling Session. In: *The Practice of Enterprise Modeling. Second IFIP WG 8.1 Working Conference, PoEM 2009* Bd. 39, 2009 (LNBIP), S. 54–68
- [Sof] SOFTWARE AG: *ARIS Produkte*, <http://www.softwareag.com/de/products/az/aris/default.asp>. – zuletzt abgerufen am 15.12.2015
- [SS14] SCHWABER, Ken ; SUTHERLAND, Jeff: *The Scrum Guide*, 2014. <http://www.scrumguides.org/scrum-guide.html>. – zuletzt abgerufen am 15.12.2015
- [Sta73] STACHOWIAK, Herbert: *Allgemeine Modelltheorie*. Springer, 1973
- [Sta06] STAUD, Josef: *Geschäftsprozessanalyse*. Springer, 2006
- [Sta15] STARKE, Gernot: *Effektive Software-Architekturen*. Hanser, 2015
- [TESW15] TILKOV, Stefan ; EIGENBRODT, Martin ; SCHREIER, Silvia ; WOLF, Oliver: *REST und HTTP. Einsatz der Architektur des Web für Integrations szenarien*. dpunkt.verlag, 2015
- [Tha12] THALHEIM, Bernhard: The Science and Art of Conceptual Modelling VI. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems* (2012), S. 76–105
- [The09] THE OPEN GROUP: *TOGAF Version 9*, 2009. <http://www.opengroup.org/architecture/togaf9-doc/arch/>. – zuletzt abgerufen am 15.12.2015
- [The13a] THE OPEN GROUP: *ArchiMate 2.1 Specification*, 2013. <http://pubs.opengroup.org/architecture/archimate2-doc/>. – zuletzt abgerufen am 15.12.2015

- [The13b] THE OPEN GROUP: *NI31 ArchiMate 2.1 Reference Card*, 2013. <https://www2.opengroup.org/ogsys/catalog/n131>. – zuletzt abgerufen am 15.12.2015
- [The13c] THE OPEN GROUP: *NI32 ArchiMate 2.1 Viewpoints Reference Card*, 2013. <https://www2.opengroup.org/ogsys/catalog/n132>. – zuletzt abgerufen am 15.12.2015
- [The13d] THE OPEN GROUP: *NI33 ArchiMate 2.1 Metamodels Reference Card*, 2013. <https://www2.opengroup.org/ogsys/catalog/n133>. – zuletzt abgerufen am 15.12.2015
- [The14] THE OPEN GROUP: *ArchiMate Tool Certification Register*, 2014. <https://archimate-cert.opengroup.org/ts-register>. – zuletzt abgerufen am 15.12.2015
- [WF06] WINTER, Robert ; FISCHER, Ronny: Essential Layers, Artifacts, and Dependencies of Enterprise Architecture. In: *EDOC Workshop on Trends in Enterprise Architecture Research* IEEE Computer Society, 2006, S. 30–38
- [Wil13] WILFING, Sabine: *Management organisationaler Anpassungsprozesse*. SpringerGabler, 2013
- [Win08] WINTER, Robert: Design Science Research in Europe. In: *European Journal of Information Systems* (2008), S. 470–475
- [Win10] WINTER, Robert: *Business Engineering Navigator. Gestaltung und Analyse von Geschäftslösungen "Business-to-IT"*. Springer, 2010
- [WPS14] WPS GMBH: *eGPM Benutzerhandbuch Version 2014.10.08 für ADONIS und die Open Models Initiative*, 2014. http://www.omilab.org/c/document_library/get_file?uuid=99d3d090-c2f1-43dc-9310-cc8792e022b8&groupId=129003. – zuletzt abgerufen am 15.12.2015
- [Zül98] ZÜLLIGHOVEN, Heinz: *Das objektorientierte Konstruktionshandbuch nach dem Werkzeug- & Material-Ansatz*. dpunkt.verlag, 1998

- [Zül05] ZÜLLIGHOVEN, Heinz: *Object-Oriented Construction Handbook*. dpunkt.verlag, 2005