

Bachelorarbeit

Tom Krause

Enterprise Architecture Management Metamodell zur
Abbildung datenintensiver Prozesse

Tom Krause

Enterprise Architecture Management Metamodell zur Abbildung datenintensiver Prozesse

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Wirtschaftsinformatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Ulrike Steffens
Zweitgutachter: Prof. Dr. Rüdiger Weißbach

Eingereicht am: 08. Dezember 2020

Tom Krause

Thema der Arbeit

Enterprise Architecture Management Metamodell zur Abbildung datenintensiver Prozesse

Stichworte

Big Data, Enterprise Architecture Management, Metamodell, ArchiMate

Kurzzusammenfassung

Diese Arbeit beschäftigt sich mit der Abbildung datenintensiver Prozesse innerhalb eines Unternehmens und der damit verbundenen Entwicklung eines Metamodells. Hierbei wird auf Basis von Unternehmensprozessen und den dazugehörigen Daten ermittelt, welche Anforderungen an die Modellierung einer Unternehmensarchitektur existieren. Dazu werden aktuelle Herausforderungen sowie Probleme aus dem Bereich Big Data analysiert, um anschließend im Kontext des Enterprise Architecture Managements ein Metamodell zu entwerfen, welches jene Voraussetzungen aufweist, um Big Data in einer Unternehmensarchitektur nutzen zu können.

Tom Krause

Title of Thesis

Enterprise architecture management metamodel for mapping data-intensive processes

Keywords

big data, enterprise architecture management, meta model, ArchiMate

Abstract

This thesis deals with the mapping of data-intensive processes within a company and the associated development of a metamodel. On the basis of company processes and the associated data, it is determined which requirements exist for the modeling of a company architecture. For this purpose, current challenges and problems in the field of big data are

analyzed in order to then design a metamodel in the context of enterprise architecture management, which has the requirements to be able to use big data in an enterprise architecture.

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel der Arbeit	1
1.3 Aufbau der Arbeit	2
2 Stand der Forschung	4
2.1 Big Data	4
2.1.1 Definition	4
2.1.2 Die 5 V's	4
2.1.3 Vorteile	7
2.1.4 Erfolgsfaktoren	8
2.1.5 Technologien	9
2.1.6 Probleme & Herausforderungen	14
2.2 Enterprise Architecture Management	19
2.2.1 Definition	20
2.2.2 Vorteile	20
2.2.3 Unterstützung für Big Data	23
2.3 Modellierung	25
2.4 Metamodellierung	29
2.4.1 Metamodell in ArchiMate	29
3 Anforderungsanalyse	31
3.1 Allgemeine Anforderungen	31
3.2 Individuelle Anforderungen	32
4 Konzeption	36

5	Umsetzung	50
5.1	Ausgangssituation	50
5.2	Erweiterung des Metamodells	53
5.2.1	Beziehungen zum Business Layer	53
5.2.2	Beziehungen zum Application Layer	54
5.2.3	Beziehungen zum Technology Layer	54
6	Evaluation	63
6.1	Modellierung von Szenarien	63
6.1.1	Anmeldung des Systems	64
6.1.2	Datenbereitstellung	65
6.1.3	Datenflüsse zu Nachbarsystemen	65
6.1.4	Pflege eines Systemverbunds	66
6.2	Bewertung	66
7	Zusammenfassung	72
7.1	Fazit	73
7.2	Ausblick	74
	Literaturverzeichnis	76
	Selbstständigkeitserklärung	82

Abbildungsverzeichnis

2.1	The 5 Vs of Big Data. Quelle: [SS17]	5
2.2	Data Management Evolution. Quelle: [SS17]	9
2.3	Datenflut: Die Zunahme der Datengröße hat die Rechenkapazitäten über- troffen. Quelle: [CZ14]	16
2.4	Knowledge discovery process. Quelle: [CZ14]	18
2.5	Traditional Enterprise Architecture Focus Areas. Quelle: [Cam14]	22
2.6	Enterprise Architecture as The Bridge between Strategy and Execution. Quelle: [Cam14]	24
2.7	ArchiMate Full Framework. Quelle: [Groa]	26
2.8	Elemente in ArchiMate Full. Quelle: [Ram19]	27
2.9	Beziehungen in ArchiMate. Quelle: [Ram19]	28
2.10	Metamodellierung an einer exemplarischen Modellsprache. Quelle: [Ram19]	29
2.11	Das Metamodell der Verhaltens- und Strukturelemente in ArchiMate. Quel- le: [Groa]	30
4.1	The meta model of concepts provided by the ISO/IEC/IEEE 42010 Stan- dard. Quelle: [ISO11]	37
4.2	Business Process Cooperation Viewpoint in ArchiMate. Quelle: In Anleh- nung an [Groa]	38
4.3	Modifizierter Business Process Cooperation Viewpoint in ArchiMate. Quel- le: Tom Krause (2020)	39
4.4	Application Usage Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]	40
4.5	Modifizierter Application Usage Viewpoint in ArchiMate. Quelle: Tom Krause (2020)	41
4.6	Layered Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]	44
4.7	Modifizierter Layered Viewpoint in ArchiMate. Quelle: Tom Krause (2020)	45

4.8	Technology Usage Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]	46
4.9	Modifizierter Technology Usage Viewpoint in ArchiMate. Quelle: Tom Krause (2020)	47
4.10	Datenkommunikation und Datendiversität durch den Data Layer. Quelle: Tom Krause (2020)	48
4.11	Medienbruch in Unternehmensprozessen. Quelle: Tom Krause (2020)	49
5.1	Erweiterung des ArchiMate Full Frameworks. Quelle: Tom Krause (2020)	51
5.2	Data Layer Metamodell. Quelle: Tom Krause (2020)	52
5.3	Elemente des Data Layers. Quelle: Tom Krause (2020)	56
5.4	Beziehungen der Metamodelle zwischen dem Business Layer und dem Application Layer. Quelle: In Anlehnung an [Groa]	57
5.5	Beziehungen der Metamodelle zwischen dem Business Layer und dem Technology Layer. Quelle: In Anlehnung an [Groa]	58
5.6	Beziehungen der Metamodelle zwischen dem Application Layer und dem Technology Layer. Quelle: In Anlehnung an [Groa]	59
5.7	Beziehungen zwischen dem Data Layer und dem Business Layer. Quelle: Tom Krause (2020)	60
5.8	Beziehungen zwischen dem Data Layer und dem Application Layer. Quelle: Tom Krause (2020)	61
5.9	Beziehungen zwischen dem Data Layer und dem Technology Layer. Quelle: Tom Krause (2020)	62
6.1	Systemübersicht. Quelle: [Gmb20]	64
6.2	Anmeldung einer Kasse. Quelle: Tom Krause (2020)	68
6.3	Datenbereitstellung für das Finanzamt. Quelle: Tom Krause (2020)	69
6.4	Datenflüsse zu Nachbarsystemen. Quelle: Tom Krause (2020)	70
6.5	Pflege des Kassensverbands. Quelle: Tom Krause (2020)	71

1 Einleitung

1.1 Motivation

Heutige Unternehmen sind abhängig von Daten, mit denen sie arbeiten müssen. Diese Daten zeichnen sich durch ihre Menge sowie Beschaffenheit aus. Die Daten stammen dabei aus diversen Quellen und haben einen Einfluss auf Geschäftsprozesse. Unternehmen, die dabei vor der Herausforderung der komplexen Datenanalyse stehen, sind über alle Industriezweige verteilt. Um einen Wert aus den großen Datenmengen zu gewinnen, sind heutige Unternehmen dazu gezwungen, Innovationen voranzutreiben. Diese Innovationen beziehen sich vor allem auf Technologien und Prozesse in Unternehmen, in denen Daten ein Bestandteil sind [MJS13].

Die riesigen Datenmengen, welche mit dem Begriff Big Data zusammengefasst werden, verursachen Veränderungen auf allen Unternehmensebenen. Diese Veränderungen implizieren den Bedarf einer Unternehmensarchitektur und deren Modellierung. Um einen Nutzen aus dieser Modellierung zu gewinnen, wird eine Methodik benötigt, die als Enterprise Architecture Management definiert wird [VBH15].

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung eines Konzepts, welches die Methodik des Enterprise Architecture Managements berücksichtigt und gleichzeitig eine Modellierung von datenintensiven Unternehmensprozessen ermöglicht. Um Vorgaben für diese Modellierungen zu erzeugen, ist ein Metamodell zu entwickeln.

1.2 Ziel der Arbeit

Das Ziel dieser Arbeit besteht folglich darin, ein Metamodell zu entwickeln, welches die Abbildung datenintensiver Unternehmensprozesse ermöglicht.

Die Basis dazu bildet ein bestehendes Metamodell einer bereits existierenden Modellierungssprache. Bei dieser weit verbreiteten Modellierungssprache handelt es sich um ArchiMate, welche von der Open Group entwickelt wurde [Groa]. Das Metamodell in ArchiMate wird dahingehend überprüft, ob es den ermittelten Anforderungen gerecht werden kann. Falls dies nicht der Fall ist, wird das Metamodell zur Integration datenintensiver Prozesse angepasst und erweitert. Dieses modifizierte Metamodell soll somit ein konzeptionelles Rahmenwerk darstellen, welches die konkrete Modellierung einer Unternehmensarchitektur gewährleistet, die für die Darstellung datenintensiver Prozesse ausgelegt ist.

Um herauszufinden, wie solch ein Metamodell aufgebaut sein muss, werden Probleme und Herausforderungen mit Big Data analysiert. Anschließend werden Anforderungen an ein Metamodell definiert, welche diesen Herausforderungen gerecht werden. Im Anschluss ist ein Metamodell zu entwerfen, das auf diesen Anforderungen basiert.

1.3 Aufbau der Arbeit

Die Arbeit ist in sieben Kapitel unterteilt. In Kapitel 2 wird der Stand der Forschung beschrieben. Hierzu gehören die Grundlagen zu den Themen Big Data (Kapitel 2.1) und Enterprise Architecture Management (Kapitel 2.2). Des Weiteren wird der Stand der Forschung zur Modellierungssprache ArchiMate (Kapitel 2.3) sowie zur Metamodellierung (Kapitel 2.4) vorgestellt.

Das Kapitel 3 beinhaltet eine Anforderungsanalyse, die die Anforderungen an das Metamodell beschreibt. Hierbei wird zwischen allgemeinen Anforderungen und individuellen Anforderungen differenziert.

Die Entwürfe, mit welchen das bestehende Metamodell in ArchiMate evaluiert wird, sind in Kapitel 4 dargestellt. Dabei wird eine Unternehmensarchitektur aus verschiedenen Sichten betrachtet. Anschließend wird dabei überprüft, ob das bereits existierende Metamodell in ArchiMate dazu geeignet ist, datenintensive Prozesse in einer Unternehmensarchitektur zu modellieren.

Das Kapitel 5 setzt die Erkenntnisse aus Kapitel 4 um. Hierbei wird das bestehende Metamodell angepasst und erweitert.

In Kapitel 6 dieser Arbeit wird das erweiterte Metamodell evaluiert. Diese Evaluation basiert auf der Modellierung von Beispielszenarien aus der Praxis. Im Anschluss wird untersucht, ob das erweiterte Metamodell dazu geeignet ist, datenintensive Prozesse abzubilden.

Das letzte Kapitel, Kapitel 7, besteht aus einer Zusammenfassung. Diese Zusammenfassung beinhaltet ein Fazit sowie einen Ausblick.

2 Stand der Forschung

Dieses Kapitel stellt die Grundlagen der Themen Big Data und Enterprise Architecture Management vor, um eine Basis für das zu entwickelnde Metamodell zu schaffen. Hierbei wird zuerst auf Erfolgsfaktoren, Technologien und zu bewältigende Probleme aus dem Bereich Big Data eingegangen. Anschließend werden Nutzen des Enterprise Architecture Managements vorgestellt sowie Probleme, die mit Big Data auftreten können. Abschließend wird auf das Konzept der Metamodellierung eingegangen, um im weiteren Verlauf der Arbeit ein Modell entwickeln zu können, welches die in diesem Kapitel erarbeiteten Grundlagen behandelt.

2.1 Big Data

2.1.1 Definition

Der Begriff definiert sich wie folgt:

"Big Data represents the Information assets characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value [DMGG15]."

2.1.2 Die 5 V's

Big Data lässt sich mittels der 5 V's charakterisieren, welche jeweils eigene Eigenschaften darstellen. Jede einzelne Eigenschaft impliziert hierbei Herausforderungen in der Verarbeitung von Daten. Hinzu kommt, dass die einzelnen Charakteristiken, wie beispielsweise die Geschwindigkeit (Velocity) und das Datenvolumen (Volume) untereinander aufgrund ihrer Merkmale konkurrieren können [SS17].

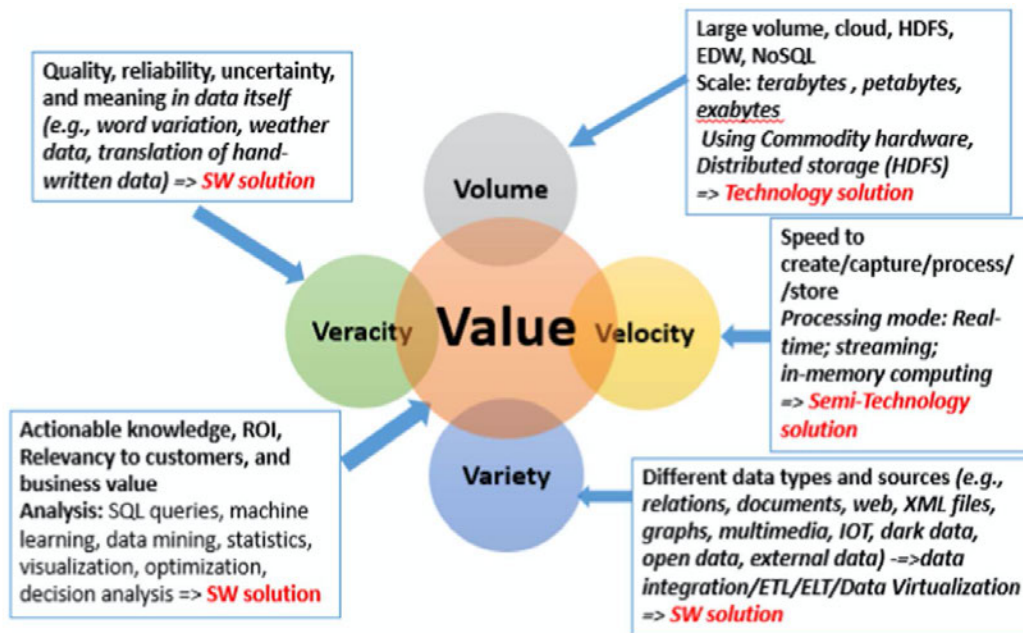


Abbildung 2.1: The 5 Vs of Big Data. Quelle: [SS17]

Volume

Die riesigen Datenmengen kennzeichnen das Volumen von Big Data [SS17]. Diese Datenmengen lassen sich anhand von Größenordnungen beschreiben. Hierbei handelt es sich um Petabyte und Exabyte von Daten [ASC17]. Folglich ist es notwendig, einen technisch-orientierten Ansatz zu finden, welcher in der Lage ist, diese Datenmengen zu verarbeiten. Dies wird z.B. durch Standardhardware sowie Hadoop verteilte Speichersysteme realisiert [SS17]. Die Tendenz dabei ist steigend, da diese Systeme stets für die Datenmengen von Big Data weiterentwickelt werden [ODS13].

Variety

Ein weiteres kennzeichnendes Merkmal von Big Data ist die Variety (Vielfalt) der Daten. Diese Datenvielfalt basiert auf verschiedenen Datentypen. Die Datenquellen können hierbei beispielsweise Sensordaten (z.B. Internet of Things) sein [GBMP13] sowie XML-Dateien oder Bilder/Videos (Multimedia). Des Weiteren kann es sich auch um Daten aus internen sowie externen Quellen handeln [SS17]. Aufgrund dieser Vielfalt stehen Unternehmen vor der Aufgabe, sich mit Daten unterschiedlicher Beschaffenheit auseinanderzusetzen, um diese vorteilhaft nutzen zu können. Diese zuvor beschriebene Vielfalt der

Daten bringt Vorteile sowie Herausforderungen mit sich [ASC17]. Um eine Datenintegration dieser Daten gewährleisten zu können, stehen Software-Lösungen im Vordergrund [SS17]. Die Abbildung 2.1 visualisiert mögliche Datenquellen sowie Datentypen.

Velocity

Die Velocity (Geschwindigkeit) der Daten beschreibt die Geschwindigkeit der Vorgänge im Zusammenhang mit der Datenverarbeitung. Dies sind die Entstehung, das Erfassen, die Extraktion, die Verarbeitung sowie die Speicherung von Daten. Um diese Vorgänge zu gewährleisten, wird eine Lösung benötigt, die semi-technologisch ist. Der technologische Teil dieses Lösungsansatzes, welcher durch eine Software realisiert werden soll, muss über Verarbeitungsmöglichkeiten verfügen, die eine Echtzeitverarbeitung von Daten ermöglichen. Des Weiteren müssen Streaming sowie In-Memory-Computing möglich sein [SS17]. Aus der Tatsache, dass es sich bei diesen Daten um Echtzeitdaten handelt, entstehen ebenfalls neue Herausforderungen. Dies ist so, da sich die aus Echtzeitdaten gewonnen Erkenntnisse und Muster jederzeit ändern können und somit bewegliche Ziele darstellen. Dies ist bei statistischen Daten nicht der Fall [ASC17].

Veracity

Ein weiteres Merkmal von Big Data ist die so genannte Veracity (Richtigkeit/Korrektheit) der Daten. Diese untersucht die Glaubwürdigkeit sowie die Zuverlässigkeit der Daten [ASC17]. Die Richtigkeit der Daten wird mittels verschiedener Fragen evaluiert. Diese Fragen thematisieren die Qualität, Unsicherheit, und Unvollständigkeit der Daten. Um die Daten aus den einzelnen Quellen bezüglich der Richtigkeit bewerten zu können, sollten einheitliche Kriterien festgelegt und benutzt werden. Ein automatisierter Ansatz zur Gewährleistung dieser Kriterien kann mittels einer Softwarelösung umgesetzt werden [SS17].

Value

Die fünfte Eigenschaft von Big Data beschreibt den Wert, welcher durch die Daten generiert wird. Dieser Wert ist jedoch nur annähernd zu beschreiben und lässt sich schwierig bestimmen. Wie in Abbildung 2.1 zu sehen ist, beschäftigen sich mögliche Maßnahmen zur Bestimmung des Werts mit der Festlegung eines Return on Investments oder auch mit der kundenbezogenen Relevanz der Daten. Um den Wert der Daten zu ermitteln, stehen verschiedene Techniken und Möglichkeiten zur Verfügung. Dies sind beispielsweise SQL-Abfragen, Data Mining, maschinelles Lernen oder auch Berechnungen aus der Statistik. Ein weiterer wichtiger Aspekt zur Bestimmung des Wertes stellt die Bereitstellung

in Form von Berichten und die Visualisierung der zuvor gewonnenen Erkenntnisse dar. Hierbei ist es vor allem wichtig, jene gewonnenen Erkenntnisse im Zusammenhang mit dem jeweiligen Geschäftsbereich zu untersuchen, um einen Geschäftswert zu generieren. Eine wichtige Rolle im Zeitalter von Big Data spielt dabei der Job des Data Scientists, welcher die Brücke zwischen der Geschäftsdomäne und der Datenanalyse darstellt, da er gleichzeitig ein Verständnis für den jeweiligen Geschäftsprozess besitzt sowie Kenntnisse über einzelne Analysewerkzeuge/-und methoden aufweisen kann [SS17].

2.1.3 Vorteile

Big Data impliziert viele Konsequenzen für Unternehmen. Diese können als Vorteile angesehen werden. Ein Kernbereich aus Big Data ist die Analytik der Datenmengen, welche eine neue Generation von Unternehmen schafft und viele Wettbewerbsvorteile mit sich bringt. Dies ist der Fall, da die Analytik einer der wesentlichen Gründe für die strategische Neuausrichtung von Unternehmen darstellt. Um diese strategischen Vorteile nutzen zu können, sind jedoch technische Investitionen sowie unternehmensweite Strategien zur Datenverwaltung notwendig [D⁺06].

Unternehmen befinden sich heutzutage in einem sich stetig ändernden Umfeld, woraus veränderte Entscheidungsprozesse resultieren. Big Data kann hierbei als Unterstützung dienen und Vorteile bieten. Zunächst einmal ermöglicht Big Data die Datenanalyse großer Datenmengen in jeglichen Geschäftsbereichen, was nicht nur Vorteile für den einzelnen Geschäftsbereich impliziert, sondern auch bereichsübergreifend in Unternehmen vorteilhaft sein kann. Dieses Potenzial, welches sich durch die Analyse der Daten ergibt, ermöglicht Unternehmen, eine präzise und bessere Entscheidungsfindung. Beispiele hierfür wären Kostenreduktionen sowie die Optimierung von Produktentwicklungen. Des Weiteren lassen sich durch die Nutzung von Big Data Unternehmensprozesse automatisieren und vereinfachen. Eine wichtige Schlüsselrolle spielt hierbei das Management eines Unternehmens, das auf Basis der gewonnenen Erkenntnisse Entscheidungen treffen muss [KP15]. Zusätzlich sind Unternehmen durch Big Data dazu fähig, neue Märkte zu erschließen [CZ14].

Weitere Vorteile für Unternehmen, die aus der Nutzung von Big-Data-Lösungen resultieren, unterscheiden sich nicht wesentlich bezüglich der Unternehmensgröße und der einzelnen Industriesektoren [Rag18]. Diese Vorteile lassen sich in Kategorien gliedern. Die erste Kategorie beschreibt so genannte Transaktionsvorteile. Beispiele hierfür wären die

Steigerung der Mitarbeiterproduktivität, Reduktion von Betriebs/-und Kommunikationskosten oder auch Einsparungen im Supply Chain Management. Eine weitere Kategorie beschreibt strategische Vorteile, wie zum Beispiel die Verbesserung von Kundenbeziehungen oder die Ausrichtung der unternehmenseigenen IT an einer Geschäftsstrategie. Zusätzlich lassen sich Transformationsvorteile mittels Big Data beschreiben. Diese äußern sich z.B. durch die Verbesserung von Geschäftsmodellen oder die Verbesserung der einzelnen Mitarbeiterfähigkeiten [JfRFWA⁺17]. Die letzte Kategorie benennt die Informationsvorteile. Beispiele hierfür sind die Verbesserung der Datengenauigkeit sowie die Möglichkeit eines einfacheren und schnelleren Datenzugriffs [GMF⁺06].

2.1.4 Erfolgsfaktoren

Um Big-Data-Projekte in Unternehmen nutzbringend durchzuführen, existieren zahlreiche Erfolgsfaktoren. Diese sind wichtig zu berücksichtigen, da immer noch rund 55 Prozent der Big-Data-Projekte in Unternehmen nicht abgeschlossen werden oder scheitern. Diese Erfolgsfaktoren lassen sich in drei Kategorien unterteilen. Die erste Kategorie thematisiert den Erfolgsfaktor Mensch. Das Berufsbild des Data Scientists spielt in dieser Kategorie eine wichtige Rolle, da dieser technisches Wissen sowie die notwendigen analytischen Fähigkeiten besitzt. Zusätzlich ist es wichtig, innerhalb von Big-Data-Projekten ein Team aus Experten zusammenzustellen, welche aus den einzelnen Geschäftsbereichen stammen, um ein diversifiziertes Wissen zu repräsentieren. Des Weiteren ist es ebenso notwendig für Unternehmen, sich in Big-Data-Projekten juristisch abzusichern. Dies soll mittels Experten und Vertretern aus dem Bereich der Compliance sowie Rechtsexperten gelingen.

Die zweite Kategorie der Erfolgsfaktoren innerhalb von Big-Data-Projekten beschreibt die Prozesse. Das Projektteam muss jederzeit in der Lage sein, auf neue Anforderungen reagieren zu können. Zusätzlich ist es wichtig, Prozesse in der Datenverarbeitung zu priorisieren. Hierbei muss das Analyseteam dazu fähig sein, die Datenqualität zu gewährleisten und zu entscheiden, ob Daten selektiert werden müssen oder nicht. Die Schwierigkeit des Teams bezüglich dieser Prozesse liegt also darin, nicht den Fokus zu verlieren. Um den Erfolgsfaktor der Prozesse messen zu können, ist es nützlich, vor Beginn von Big-Data-Projekten Geschäftsziele zu definieren, beispielsweise mittels einer Kosten-Nutzen-Analyse. Damit die Analyseprozesse erfolgreich werden können, sollte iterativ innerhalb der Projektteams gearbeitet werden.

Ein weiterer Erfolgsfaktor sind die eingesetzten Technologien in Big-Data-Projekten. Dabei ist es wichtig, sich auf mehrere Technologien zu konzentrieren. Bezüglich der Technologien treten häufig Probleme auf. Eines dieser Probleme ist die Nutzung von universellen Softwarearchitekturen, weshalb die eingesetzte Technologie stets individuell in Bezug auf das Projektziel erstellt werden sollte. Ein weiteres Beispiel für den Erfolgsfaktor Technologie ist das Problem der Datenspeicherung, wobei neue Technologien benötigt werden, da die herkömmlichen Datenbanken aufgrund ihres stark strukturierten Aufbaus (Tabelleform) den Anforderungen nicht mehr genügen [GSK15].

2.1.5 Technologien

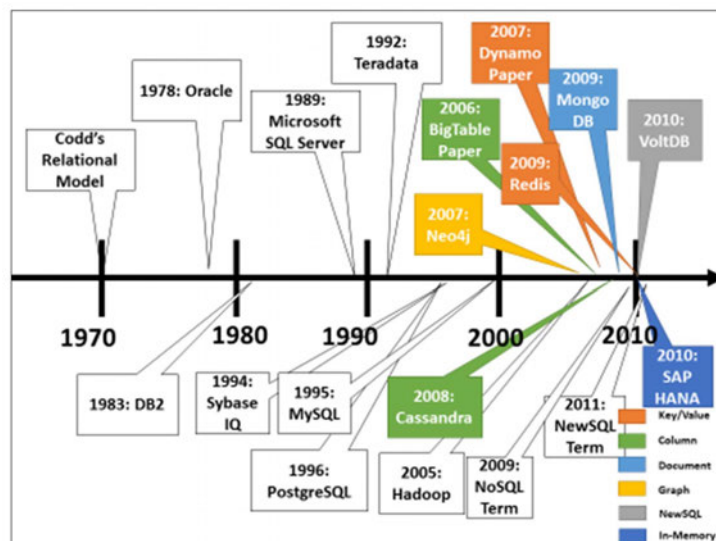


Abbildung 2.2: Data Management Evolution. Quelle: [SS17]

Im Laufe der Jahre haben sich die Technologien der Datenbanken an die Bedürfnisse von Big Data angepasst. Abbildung 2.2 stellt den Zeitverlauf der Entwicklung der Datenbanktechnologien dar. Hierbei ist zu erkennen, dass ab 1970 relationale Datenbanken verwendet wurden. Die relationale Datenspeicherung basiert auf Tupeln, welche die Daten in Tabellen abbilden. Bei diesen Tupeln handelt es sich um Beziehungen zwischen den Daten [SS17]. Des Weiteren stellen Tupel eine Menge von Attributen dar [MU03].

Aufgrund der immer größer werdenden Datenmengen sowie der Heterogenität der Daten, reichten die relationalen Datenbanken nicht mehr aus. Dafür gibt es diverse Gründe. Erstens sind die Datenmengen vielfältig, zu groß und müssen schnell verarbeitet werden.

Des Weiteren ist das Schema dieser Datenbanken, welches beim Speichern der Daten in der Datenbank zum Einsatz kommt, zu starr. Ein Grund dafür sind die zu strengen ACID-Eigenschaften. ACID steht für Atomicity (Atomarität), Consistency (Konsistenz), Isolation (Abgrenzung) und Durability (Dauerhaftigkeit). Die relationalen Datenbanken werden aber weiterhin für transaktionale Anwendungen genutzt [SS17].

Die Skalierung der Systeme, welche sich durch die ACID-Eigenschaften auszeichnen, gestaltet sich schwierig. Diese Schwierigkeit basiert auf dem CAP-Theorem. CAP steht hierbei für Consistency (Konsistenz), Availability (Verfügbarkeit) und Partition Tolerance (Ausfalltoleranz). Das CAP-Theorem besagt, dass es in verteilten Systemen nicht möglich ist, gleichzeitig die drei zuvor beschriebenen Eigenschaften (CAP) zu gewährleisten [Bre00]. Dies ist der Fall, da in verteilten Systemen eine Ausfalltoleranz gegeben sein muss. Hierbei ist jedoch die Konsistenz nicht mehr gewährleistet. Infolgedessen sind Systemarchitekturen notwendig, welche auf den so genannten BASE-Eigenschaften basieren [MH13]. BASE wird vorwiegend in NoSQL-Systemen gewährleistet und steht für Basically Available (Grundsätzlich verfügbar), Soft State, Eventual Consistency.

Des Weiteren zeigt die Abbildung 2.2 die Entwicklung von Systemen, welche intern von der Struktur der relationalen Datenbanken abweichen. Ein Beispiel hierfür ist das Hadoop-Framework, welches in der Lage ist, auch unstrukturierte Daten zu archivieren. Hadoop ist intern mittels des MapReduce Konzepts realisiert, welches noch im Verlauf dieser Arbeit genauer erläutert wird. Des Weiteren zeigt Abbildung 2.2, dass ab 2010 NoSQL- und NewSQL-Datenbanken eingesetzt wurden, welche ebenfalls noch im Verlauf dieser Arbeit vorgestellt werden. Diese Datenbanken ermöglichen Skalierbarkeit mittels einer Scale-Out-Architektur [SS17].

Hadoop

Apache Hadoop ist ein Open-Source-Framework, welches den MapReduce-Algorithmus umsetzt [SS17]. Dieser Algorithmus besteht aus zwei Schritten, Map und Reduce. Der erste Schritt des Algorithmus erhält als Input eine Menge von Daten und selektiert aus dieser Menge die benötigten Daten für den nächsten Schritt. Diese werden dann als Schlüssel-Wert-Paare zusammengefasst. Im nächsten Schritt, dem Reduce-Schritt, wird das Ergebnis des Map-Schritts als Input genommen. Dabei handelt es sich um die gebildeten Tupel, welche zu einer kleineren Tupelmengge aggregiert werden. Hierbei werden die Ergebnisse mit gleichem Schlüssel aus den Schlüssel-Wert-Paaren zusammengefasst [DG08]. Des Weiteren handelt es sich bei Apache Hadoop um ein Framework, welches in der Lage ist, unstrukturierte Daten miteinander zu kombinieren und somit als Grundlage

für Data Mining dient. Die Datenspeicherung des Frameworks basiert auf einem HDFS (Hadoop Distributed File System). Dieses Dateisystem weist jedoch die Schwachstelle auf, dass eine manuelle Programmierung bei zu vielen Rechenoperationen notwendig ist. Folglich wurde das Spark-Framework entwickelt, das jene Schwachstelle in der Datenspeicherung beseitigt [SS17].

Apache Spark

Bei Apache Spark handelt es sich um ein Open-Source-Framework, das seinen Ursprung an der University of California in Berkeley fand. Spark ist darauf ausgelegt, schnelle Datenabfragen auf Daten zu ermöglichen, welche sich durch ihre große Menge und verschiedene Quellen kennzeichnen. Technisch realisiert wird dies mittels einer verteilten Architektur sowie Cluster Computing. Die Architektur des Frameworks besteht aus verschiedenen Komponenten. Diese Komponenten sind Spark Core, Spark SQL, Spark Streaming, MLlib Machine Learning Library und GraphX. Der Kern des Systems (Spark-Core) ist für diverse Aufgaben innerhalb des Systems zuständig: Scheduling, Steuerung der Ein/-Ausgabeprozesse, Aufgabenverteilung. Die Struktur der Daten im Kern ist technisch durch RDDs (Resilient Distributed Datasets) realisiert. Diese zeichnen sich dadurch aus, dass sie die Verarbeitung auf mehreren Rechnern ermöglichen. Die Umwandlung dieser RDDs findet durch die zweite Komponente des Spark-Frameworks, Spark SQL, statt. Diese Umwandlung basiert darauf, dass aus den RDDs Data Frames erzeugt werden, um anschließend SQL-Abfragen durchführen zu können. Des Weiteren zeichnet sich Apache Spark dadurch aus, dass es Algorithmen für das maschinelle Lernen zur Verfügung stellt. Dies geschieht intern durch die MLlib Machine Learning Library. Die Aufgabe der GraphX-Komponente besteht darin, ein Framework zur Erzeugung von Graphen bereitzustellen. Zusätzlich zeichnet sich Apache Spark durch seine Performance sowie Skalierbarkeit der Leistungsfähigkeit durch Spark Cluster aus [SDC⁺16].

NoSQL- und NewSQL-Datenbanken

Bei den NoSQL-Datenbanken handelt es sich um Speichersysteme, die auf einem nicht-relationalen Schema basieren. Diese sind intern durch Schlüsselwertspeicher, Spaltenspeicher, Dokumentenspeicher und Diagrammdatenbanken realisiert [SDC⁺16, SF13]. Der Vorteil der NoSQL-Datenbanken besteht im Gegensatz zu relationalen Datenbanken darin, dass sie bezüglich der Konsistenz weniger Beschränkungen aufweisen und somit keine festen Tabellenschemata besitzen. Folglich existieren für verschiedene Anwendungen individuelle Datenmodelle. Das CAP-Theorem spielt bei diesen Datenbanken ebenfalls eine Rolle, da es anders ausgeprägt ist. Bei No-SQL-Datenbanken werden die Verfügbarkeit

(Availability) und Ausfalltoleranz (Partition Tolerance) angestrebt. Das CAP-Theorem steht zusätzlich für Consistency (Konsistenz) [SS17].

Bei NewSQL-Datenbanken handelt es sich um neuartige Datenbanken, welche die Stärken der relationalen Datenbanken mit den Stärken der NoSQL-Datenbanken kombinieren. Folglich ermöglichen sie die SQL Nutzung, gewährleisten die bereits zuvor beschriebenen ACID-Eigenschaften und sind mittels einer Scale-Out-Architektur aufgebaut [Sto12].

In-Memory Computing

In-Memory-Datenbanken verfolgen einen anderen Ansatz der Datenspeicherung. Hier werden alle Daten im RAM gespeichert und nicht auf der Festplatte. Diese Datenbanken sind relational und ermöglichen somit die Nutzung von SQL. Die Persistenz dieser Datenbanken wird durch Momentaufnahmen (Snapshots) sowie durch Protokolle der einzelnen Transaktionen gewährleistet.

Hybrid Cloud Computing

Beim Cloud Computing handelt es sich um einen Pool aus gemeinsam nutzbaren IT-Ressourcen, welche über Service Provider zur Verfügung gestellt werden. Bei diesen Ressourcen handelt es sich beispielsweise um Netzwerke, Speicher, Server, Anwendungen und andere Services. Kennzeichnend für das Cloud-Modell sind einzelne Merkmale der Cloud sowie die einzelnen Servicemodelle und Bereitstellungsmodelle. Bei den Merkmalen der Cloud handelt es sich um Selbstbedienung nach eigenem Bedarf und auf Abruf, breiten Netzwerkzugang, hohe Anzahl an Ressourcen sowie Elastizität. Grundsätzlich lässt sich das Cloud Computing in drei Servicemodelle unterteilen: Software as a Service (SaaS), Platform as a Service (PaaS) und Infrastructure as a Service (IaaS). Des Weiteren existieren vier Bereitstellungsmodelle des Cloud Computings: Hybrid Cloud, Private Cloud, Community Cloud und Public Cloud [SS17].

Das Servicemodell Infrastructure as a Service dient vor allem dazu, individuelle und virtuelle Cloud-Lösungen zusammenzustellen. Bei den Ressourcen handelt es sich z.B. um Netzwerke, Speicher sowie um andere Computerressourcen. Hierbei steht der Nutzer der Cloud vor der Aufgabe, die richtige Software auszuführen und die Funktionalität der Software zu gewährleisten. Die Cloud selbst wird hierbei nicht vom Nutzer verwaltet, der Nutzer besitzt jedoch die Kontrolle über die eingesetzten Ressourcen. Mittels des Servicemodells Platform as a Service hat der Nutzer der Cloud die Möglichkeit, Werkzeuge und Services für die Entwicklung und Inbetriebnahme für Anwendungen zu nutzen.

Der Nutzer selbst ist dabei nicht befugt, die Infrastruktur der Cloud zu verwalten, jedoch darf er die zur Verfügung gestellten Ressourcen (Anwendungen) kontrollieren. Das letzte Servicemodell Software as a Service dient dazu, einen Zugang zu diversen Anwendungsprogrammen sowie Software zu schaffen. Diese Software wird dann individuell für die benötigten Anforderungen in der Cloud ausgewählt und ausgeführt. Der Zugriff auf diese Software erfolgt über Schnittstellen. Hierbei besitzt der Nutzer ebenfalls nicht die Möglichkeit, die Infrastruktur der Cloud zu verwalten. In Ausnahmefällen ist der Nutzer jedoch dazu berechtigt, die Anwendungen konfigurieren zu können [MG⁺11].

Bezüglich der Bereitstellungsmodelle im Cloud Computing, ist die hybride Form der Cloud ein häufig verwendetes Modell. Es handelt sich hierbei um eine Mischung aus einer öffentlichen sowie privaten Cloud. Die hybride Cloud hat den Vorteil, dass basierend auf den jeweiligen Anforderungen der Nutzer, die Arbeitslasten zwischen öffentlichen und privaten Clouds wechseln können. Des Weiteren findet in der hybriden Cloud eine Trennung der Datenspeicherung statt. Geschützte Daten werden in der privaten Cloud gespeichert und Daten, welche nicht als private Daten zu schützen sind, innerhalb der öffentlichen Cloud. Aufgrund dessen sind Nutzer in der Lage, je nach Arbeitslast, private Daten schnell zu erhalten sowie die Vorteile einer öffentlichen Cloud zu nutzen. Folglich erhalten Unternehmen mehr Flexibilität im Umgang mit Cloud-Lösungen. Allgemeine Vorteile einer Cloud-Nutzung in Unternehmen bestehen zusätzlich in einer hohen Skalierbarkeit und einer einfachen Wartung der Cloud. Andererseits stehen Unternehmen vor der Herausforderung, die Daten innerhalb der Cloud gut zu verwalten und korrekt zu integrieren.

ETL & ELT

ETL (Extract, Transform & Load) steht für extrahieren, transformieren und laden der Daten. Bei ELT (Extract, Load & Transform) handelt es sich um extrahieren, laden und transformieren der Daten. Hierbei wird der Unterschied zwischen beiden Techniken deutlich. Beim ETL werden die Daten nach der Extraktion aus der Datenquelle bereits in der Vorverarbeitung transformiert bevor sie endgültig auf einem Server gespeichert werden. Diese Technik ist jedoch aufgrund der großen Datenmengen von Big Data nicht mehr ausreichend. Folglich wird ELT verwendet, wobei die Transformation der Daten erst in der endgültigen Zieldatenbank stattfindet nachdem die Daten aus der Quelle extrahiert wurden. Der Vorteil hierbei besteht darin, Daten erst zur Laufzeit zu bearbeiten, wenn diese benötigt werden. Dies ermöglicht die Erstellung von so genannten Data Marts sowie die Erstellung von Plattformen zur Datenintegration, welche Echtzeitanalysen ermögli-

chen. Die Daten, welche hierbei bearbeitet und verwendet werden, sind zum Beispiel nützlich für das maschinelle Lernen oder zur Erstellung von Dashboards [SS17].

2.1.6 Probleme & Herausforderungen

Strategische Herausforderungen

Die erste Herausforderung, vor der Unternehmen durch Big Data stehen, ist die Frage der strategischen Ausrichtung seitens der Unternehmensführung. Durch die riesigen Datenmengen und die sich daraus ergebenden Möglichkeiten, besitzt Big Data eine Relevanz für die Managementstrategie der heutigen Firmen [CK15]. Folglich sind strategische Neuausrichtungen dieser Unternehmen die Konsequenz. Hierbei kann es sich um Marketingstrategien handeln, wie zum Beispiel die Erstellung von Kundenprofilen [Dav14]. Des Weiteren besteht eine strategische Herausforderung in der Erschließung neuer Geschäftsfelder/-bereiche. Diese kann beispielsweise durch Techniken wie Data Mining vollzogen werden [Cle09, CCS12]. Um Techniken wie das Data Mining durchführen zu können, spielen Daten aus dem internen sowie externen Umfeld des Unternehmens eine wichtige Rolle, da mittels dieser Daten neue Muster und Trends gewonnen werden können [D⁺06, BC12, Dav14]. Eine weitere Schwierigkeit basiert darauf, dass langfristige Vorhersagen für die Entwicklung von Mustern und Trends nicht möglich sind. Folglich sind jene Tools, die auf langfristige Vorhersagen ausgerichtet sind, nicht aussagekräftig, da ständig neue Daten gewonnen werden und sich die Beziehungen/-Abhängigkeiten zwischen diesen Daten jederzeit ändern. Somit zeigt sich, dass eine weitere Herausforderung in der Arbeit mit Big Data in dem kurzen Zeithorizont der Aussagekraft der Daten liegt. Demnach stehen heutige Unternehmen vor der Aufgabe, in Echtzeit die Datenmengen zu verarbeiten und Aussagen über diese Daten zu treffen [CK15].

Zusätzlich ist es wichtig, ein Enterprise Information Management innerhalb von Unternehmen aufzubauen, um die notwendige Infrastruktur für Big Data zu besitzen. Das Enterprise Information Management regelt jegliche Fragestellungen im Zusammenhang mit der Datenverwaltung, Datenintegration und der Datenqualität [Kal12].

Weiterhin ist es wichtig, die einzelnen Zeitabschnitte in Big-Data-Projekten zu priorisieren. Der wichtigste Abschnitt ist die Zeit von der Festlegung des zu analysierenden Problems bis zum Erhalt der Informationen in Echtzeit, um nicht alle Daten verarbeiten zu müssen. Folglich stehen Unternehmen auch hier vor der Herausforderung zu bestimmen, wie viele Daten für diesen Zeitabschnitt in Big-Data-Projekten notwendig sind, um

anschließend die richtige Systemarchitektur zu erstellen sowie die passenden Algorithmen zur Datenverarbeitung zu implementieren [KAEM13].

Ebenso existieren diverse weitere Herausforderungen für Big-Data-Projekte. Eine Herausforderung hierbei besteht zum Beispiel darin, die Geschäftsziele zu identifizieren, um anschließend die individuellen Anforderungen einzelner Big-Data-Projekte festzulegen. Darauf aufbauend ist es wichtig, einzelne Anwendungsfälle des Projekts zu ermitteln, damit die benötigte Systemarchitektur erstellt werden kann. Die Governance (Führung) in Projekten spielt hierbei eine wichtige Rolle. Um diesen Herausforderungen gerecht zu werden, kann eine konzeptionelle Modellierung helfen. Diese ermöglicht eine technische Perspektive für Big-Data-Projekte sowie eine Managementsicht, um gegenseitiges Verständnis zwischen Analysten und der Unternehmensführung zu ermöglichen.

Wie bereits in Kapitel 2.1.2 beschrieben, sind in Abbildung 2.1 die 5 V's dargestellt, welche Big Data kennzeichnen. Diese 5 V's stellen ebenfalls Herausforderungen für Projekte dar, die sich mittels konzeptioneller Modellierung lindern lassen. Für das Volume (Volumen) kann eine Modellierung wichtige Daten und Metadaten identifizieren und beschreiben. Dies ist ebenfalls nützlich, um der Velocity (Geschwindigkeit) der Daten gerecht zu werden, da mittels der Modellierung wichtige Daten gefiltert werden können. Bezüglich der Veracity (Richtigkeit) der Daten hilft eine Modellierung dabei, die Qualität und Vollständigkeit der Daten zu prüfen. Zusätzlich ist ein konzeptionelles Modell in der Lage, den Value (Wert) eines Big Data Projekts zu analysieren und Ergebnisse zu interpretieren. Inwiefern eine Modellierung der Unternehmensarchitektur als Unterstützung für Big Data dienen kann, wird in Kapitel 2.2.3 erläutert.

Eine weitere Herausforderung in Projekten stellt die Rekrutierung qualifizierter Arbeitskräfte mit analytischem Denkvermögen dar. Ein Beispiel hierfür wäre der Data Scientist. Zudem ist es wichtig, die einzelnen Big-Data-Konzeptionen zu verstehen, wie beispielsweise Architektur-Lösungen, die auf Cloud-Computing oder Spark-Frameworks mit Parallelverarbeitung basieren [SS17].

Des Weiteren lassen sich Probleme in dem Zugang zu Daten sowie in der Kommunikation zwischen Stakeholdern nachweisen. Hierbei besteht die Schwierigkeit darin, externe und interne Daten in eine gemeinsame Plattform zur Analyse zu integrieren. Die Ursache dafür liegt darin, dass nicht alle Daten aus einzelnen Abteilungen des Unternehmens unternehmensweit zugänglich sind. Der Grund dafür sind so genannte Informationssilos in einzelnen Unternehmensbereichen. Folglich gestaltet sich die Kommunikation zwischen

den Stakeholdern als schwierig, da das Vertrauen in Teile der Daten seitens der Stakeholder fehlt. Dies ist der Fall, da die einzelnen Bereiche in Unternehmen sich nur um das Lösen der eigenen individuellen Probleme mittels der Daten kümmern [Cam14].

Tools und Technologien

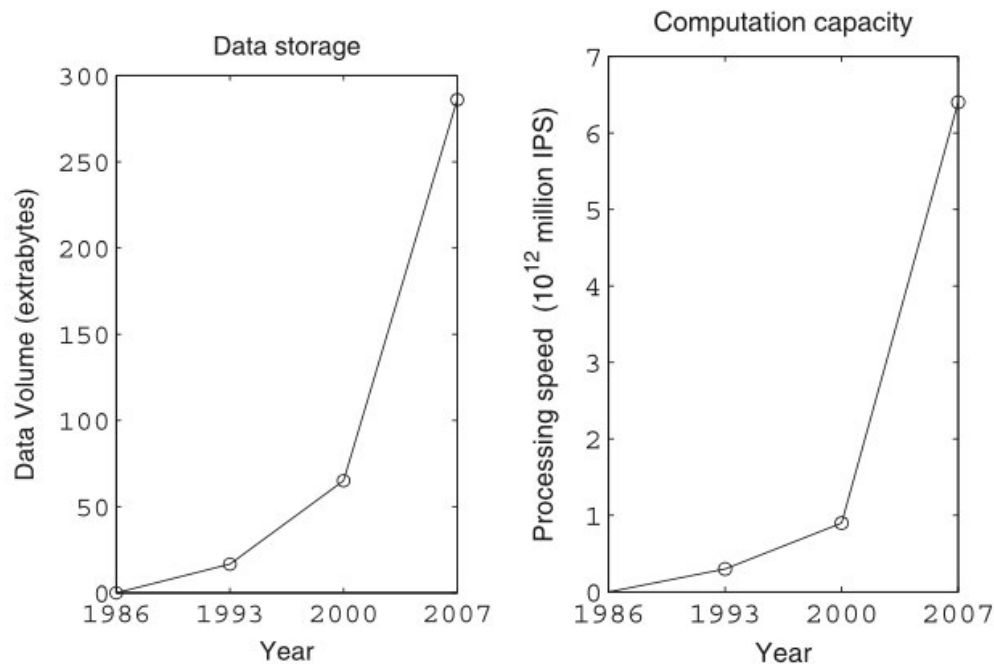


Abbildung 2.3: Datenflut: Die Zunahme der Datengröße hat die Rechenkapazitäten übertrifft. Quelle: [CZ14]

Die nächste Herausforderung ist technischer Natur. Hierbei stehen die Infrastruktur für die Arbeit mit Big Data sowie die notwendigen Technologien im Vordergrund. Bezüglich der richtigen Infrastruktur existieren verschiedene Prämissen. Die erste Voraussetzung besteht in einem Big-Data-Ökosystem. Hierzu gehören die richtige Hardware (Rechner, NoSQL- Datenbanken), Unternehmensarchitekturen sowie die richtigen Frameworks, wie beispielsweise das Hadoop-Framework [Kal12]. Im Zusammenhang mit den riesigen Datenmengen stellt die technische Infrastruktur eine Herausforderung dar, da die standardmäßigen Technologien zur Datenverarbeitung nicht mehr ausreichen [SS17]. Der überproportionale Anstieg der Datenmengen im Vergleich zu den verfügbaren Technologien ist in Abbildung 2.3 dargestellt. Die Bereitstellung geeigneter Technologien, wie z.B. durch das Cloud Computing, ist im Kapitel 2.1.5 beschrieben.

Weiterhin ist es wichtig, die richtigen Data Science Tools zu nutzen. Diese Tools müssen folgendes enthalten: Mittel zur Datenextraktion/ und -visualisierung sowie Algorithmen für maschinelles Lernen. Zusätzlich müssen Techniken aus dem Bereich der Statistik anwendbar sein sowie eine Berichterstattung mittels Scorecards oder Dashboards [Kal12].

Probleme bezüglich des Umgangs mit Daten

Grundsätzlich lassen sich weitere Herausforderungen und Probleme im Zusammenhang mit Big-Data-Projekten in drei Bereiche unterteilen. Dies sind Speicherprobleme, Verwaltungsprobleme sowie Verarbeitungsprobleme. Zusätzlich stehen Unternehmen bezüglich des Managements von Big-Data-Projekten vor der Herausforderung, neue Methoden zur Datenvalidierung zu entwickeln.

Des Weiteren existieren so genannte dynamische Designprobleme. Hierbei wird der Output-Prozess der Daten thematisiert. Der Output-Prozess ist jener Prozess, der nach der Datenextraktion stattfindet und somit die Datenverarbeitung behandelt. Auf Basis dieses Prozesses können die Ergebnisse der Datenanalyse gewonnen werden. Dieser Output-Prozess stellt eine noch größere Herausforderung dar als der Input-Prozess. Der Input-Prozess beschreibt den Zustand der Daten während des Zugriffs auf die Daten aus der Datenquelle, bevor die Daten bearbeitet werden. Dieser Zugriff auf riesige Daten, welche halbstrukturiert sowie unstrukturiert sind, kann sich ebenfalls als problematisch erweisen.

Weitere Herausforderungen bezüglich Big Data, vor denen Unternehmen stehen, sind einzelne Abwägungen, die innerhalb von Projekten durchzuführen sind. Die erste Abwägung ist jene, welche die Datenqualität der Datenquantität gegenüberstellt. Hierbei ist die Entscheidung zu treffen, welche Daten für ein Projekt relevant oder irrelevant sind. Zusätzlich stellt sich die Frage, wie man den Wert der Daten im Prozess der Entscheidungsfindung bestimmt. Weiterhin kann es zu einer zusätzlichen Herausforderung kommen, welche das Wachstum der Daten thematisiert. Dies ist so, da die Datenbestände in Unternehmen stetig wachsen, wie beispielsweise Kundendaten oder Projektdaten.

Außerdem ist die Verbreitung von Daten ein Problem. Dies ist der Fall, da die Middleware zur Kommunikation in Firmen ein Problem darstellt, da diese im Vergleich zur Kommunikationshardware nicht ausreichende Geschwindigkeiten liefert. Folglich werden die Daten zu langsam transportiert.

Zusätzlich stellt die Arbeit mit Personendaten eine große Herausforderung dar, da oftmals hierbei eine Verletzung der Privatsphäre stattfinden könnte. Aus diesem Grund stellt die Vertraulichkeit der Personendaten ein Problem dar.

Um mit den riesigen Datenmengen arbeiten zu können, ist eine verteilte Verarbeitung der Daten notwendig. Eine Replikation der Hardware und eine Erweiterbarkeit des Systems (Cloud Computing und parallele Programmierung mittels MapReduce) stellen dafür Lösungsansätze dar. Die Herausforderung hierbei liegt jedoch in der Verarbeitungszeit der Daten, da die einzelnen Knoten miteinander kommunizieren müssen. Folglich stellt diese Kommunikation ein Effizienzproblem dar [KAEM13].

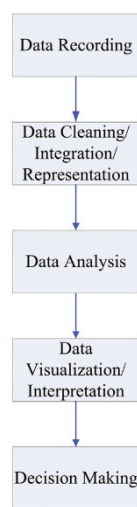


Abbildung 2.4: Knowledge discovery process. Quelle: [CZ14]

Weitere Herausforderungen ergeben sich aus der Speicherung, Suche, Analyse, Freigabe und Visualisierung von Daten. Außerdem existieren diverse Herausforderungen im Analyseprozess, welcher in Abbildung 2.4 dargestellt ist. Die Probleme und Herausforderungen entstehen hierbei aus den einzelnen Phasen des Analyseprozesses [CZ14]. Diese liegen unter anderem in der Dateninkonsistenz, Skalierbarkeit und Aktualität der Daten [ABB⁺11, KAE⁺09]. Aufgrund der großen Datenmengen sowie der Datenbeschaffenheit sind Unternehmen zusätzlich dazu gezwungen, so genannte Vorverarbeitungstechniken für die Daten zu nutzen. Dies können Techniken wie Datenbereinigungen sowie Datentransformationen sein [HPK11]. Um die einzelnen Herausforderungen der Teilprozesse zu evaluieren, ist es nützlich, die jeweiligen Prozesse einzeln zu betrachten.

Aufgrund der stetig steigenden Größe der Datensätze, werden Innovationen bezüglich der Datenspeicherung benötigt, da die bestehenden Speicherarchitekturen den Anforderungen nicht mehr genügen. Bei diesen bestehenden Architekturen handelt es sich um Direct-Attached Storage (DAS), Network-Attached Storage (NAS) sowie Storage Area Network (SAN) - Systeme. Diese Architekturen genügen den heutigen Anforderungen nicht mehr, da die Parallelität und der Durchsatz pro Server nicht gewährleistet sind. Aufgrund der riesigen Datensätze, welche aus Big Data resultieren, ist ein Datenmanagement für die Verwaltung dieser Daten erforderlich, welches im Kern aus NoSQL-Datenbanken besteht. Beispiele hierfür sind Apache Cassandra oder Apache HBase. NoSQL-Datenbanken eignen sich für die Verwaltung von Big Data, da diese schemafrei sind, Replikation unterstützen und gute Schnittstellen besitzen. Dies wird genauer in Kapitel 2.1.5 beschrieben. Um die parallele Datenverteilung auf verschiedenen Knoten zu ermöglichen, wird das MapReduce-Prinzip eingesetzt [CZ14].

Zusätzliche Schwierigkeiten existieren in der Datenerfassung, da die Daten aus verschiedenen Quellen stammen [RMRESC⁺16]. Bei diesen Quellen handelt es sich beispielsweise um Transaktionsdaten [MCB⁺11], Metadaten [GR11], Daten aus sozialen Medien [Man11, BB12, MS13] oder Sensordaten [DX14, LGPA⁺13]. Bezüglich der Daten existieren ebenfalls Schwierigkeiten für Unternehmen. Hierbei handelt es sich um die Übertragung der Daten [HWCL14] aus den einzelnen Quellen und die Datenvorverarbeitung [HWCL14, ABB⁺11].

Darüber hinaus stehen Unternehmen, welche Big Data nutzen wollen, vor der Herausforderung der Datenvisualisierung [Sut14]. Die Visualisierung der Daten eignet sich vor allem für statistische Analysen sowie zur Betrachtung einzelner Daten im Gesamtkontext. Des Weiteren lassen sich durch die Visualisierung Muster identifizieren sowie Korrelationen zwischen Daten darstellen. Um die Datenvisualisierungen effizient nutzen zu können, sollten diese interaktiv für die jeweiligen Datensätze generiert werden können [FD12].

2.2 Enterprise Architecture Management

In diesem Kapitel werden die Grundlagen zum Enterprise Architecture Management sowie dessen Vorteile beschrieben. Das Ziel dieser Arbeit besteht darin, ein Enterprise Architecture Management Metamodell für datenintensive Prozesse zu entwerfen. Folglich

ist es notwendig, die Unterstützung einer Unternehmensarchitektur sowie deren Management für Big Data zu beschreiben.

2.2.1 Definition

Enterprise Architecture

Eine Definition lautet wie folgt:

"the description of the stakeholders mission including information, functionality, location, organization, and performance parameters. Enterprise Architecture describes the plan for building a system or set of systems [Osv01]."

Enterprise Architecture Management

Nach Ernst et al. ist Enterprise Architecture Management:

"[...] a continuous and iterative process controlling and improving the existing and planned information technology (IT) support for an organization. The process not only considers the IT of the enterprise, but also business processes, business goals, strategies, etc. are considered in order to build a holistic and integrated view on the enterprise.

The goal is a common vision regarding the status quo of business and IT as well as of opportunities and problems arising from these fields, used as a basis for a continually aligned steering of IT and business [ELSW06]."

2.2.2 Vorteile

Vorteile einer Unternehmensarchitektur

Im ersten Teil dieses Kapitels werden zunächst allgemeine Vorteile einer Unternehmensarchitektur dargestellt, um anschließend die Vorteile und den Nutzen eines Enterprise Architecture Managements zu erläutern.

Unternehmen stehen heutzutage vor der Aufgabe, sich mit einem stetig wachsenden Umfeld zu befassen. Dieses Umfeld kennzeichnet sich durch stetige Veränderungen. Hierzu

gehört z.B. die Einführung von neuen Technologien. Zusätzlich basieren diese Veränderungen auf Fusionen, Übernahmen sowie dem globalen Wettbewerb zwischen Unternehmen. Folglich bedarf es stets neuer Innovationen in Unternehmen, um wettbewerbsfähig zu bleiben und den unternehmenseigenen Marktwert zu erhalten. Das Management der Unternehmen steht hierbei vor der Aufgabe, Lösungen zu finden, um die zuvor beschriebenen Herausforderungen zu bewältigen. Eine Unternehmensarchitektur und deren Management hilft Unternehmern hierbei. Die Unternehmensarchitektur dient ebenso dazu, einen Gesamtüberblick über das Unternehmen zu erhalten, um schnell auf gesellschaftliche und technologische Veränderungen reagieren zu können [OLPW⁺08].

Des Weiteren kann eine Unternehmensarchitektur dabei helfen, Geschäftsvisionen für Unternehmen zu definieren, um diese anschließend mittels einer Darstellung der notwendigen Mittel (Anforderungen, Modelle etc.) zu verwirklichen. Um diese zukünftigen Veränderungen in Unternehmen zu realisieren, sind mehrere Komponenten beteiligt. Bei diesen Komponenten handelt es sich um Personen, Prozesse, Informationen und Technologien eines Unternehmens sowie deren Beziehungen untereinander [LAB⁺08]. Der Aufbau einer Unternehmensarchitektur und das damit verbundene Management dienen dazu, eine Übersicht über diese Komponenten zu erhalten und einzelne Phasen im Entwicklungsprozess des Unternehmens zu visualisieren. Somit ist die Unternehmensarchitektur eine Unterstützung für Unternehmen in Form einer Landkarte, welche den notwendigen Verlauf der Entwicklung darstellt, um derzeitige Lücken in Unternehmen zu schließen [GP11]. Aufgrund der Darstellung dieses Entwicklungsverlaufs dient eine Unternehmensarchitektur ebenfalls dazu, innerhalb der einzelnen Phasen eine bessere Entscheidungsfindung zu ermöglichen und diese Entscheidungen im Sinne der Unternehmensstrategie zu tätigen. Folglich dient eine Architektur zur Verbesserung der Effektivität, Effizienz sowie Agilität in Unternehmen. Der Großteil der heutigen Unternehmensarchitekturen konzentriert sich auf die IT-Ressourcen. Dementsprechend stellen die Unternehmensarchitekturen eine Grundlage für die Planung, Architektur und Weiterentwicklung der IT-Systeme dar.

Ein weiterer Aspekt stellt die Aufteilung der Unternehmensarchitekturen in vier Schwerpunkte dar. Dies ist somit ein weiterer Vorteil, da Unternehmen durch diese Schwerpunkte und deren Verbindung aus unterschiedlichen Perspektiven genauer beschrieben werden können. Um die Vorteile einer Unternehmensarchitektur hinsichtlich der einzelnen Schwerpunkte darzustellen, ist es wichtig, die einzelnen Schwerpunkte zunächst aufzulisten. Der erste Bereich behandelt die einzelnen Komponenten, die bezüglich des Geschäfts innerhalb einer Unternehmensarchitektur visualisiert werden. Die Geschäftsebene stellt unter anderem folgende Komponenten dar: Strategiekarten, Unternehmensziele, Unter-

nehmensrichtlinien, Geschäftsprozesse, Lieferanten (Hardware, Software, Dienstleistungen). Die zweite Ebene visualisiert die Anwendungen. Hierbei werden unter anderem Software, Schnittstellen und Datenströme aufgezeigt. Bei der dritten Ebene in Unternehmensarchitekturen handelt es sich um die Schicht der Informationen. Diese visualisiert beispielsweise die Metadaten und Daten eines Unternehmens. Zusätzlich werden Datenmodelle, Beziehungen zwischen Daten sowie einzelne Geschäftsdomänen dargestellt. Die vierte Ebene, die Technologieebene, visualisiert die technische Infrastruktur eines Unternehmens. Beispiele hierfür wären die Hardware, Server, Betriebssysteme und Datenbanksysteme. Aufgrund dieser einzelnen Schwerpunkte, dient eine Unternehmensarchitektur dazu, jegliche Beziehungen und Abhängigkeiten zu steuern. Die Abbildung 2.5 stellt die Beziehungen zwischen den einzelnen Schichten der Unternehmensarchitektur dar.

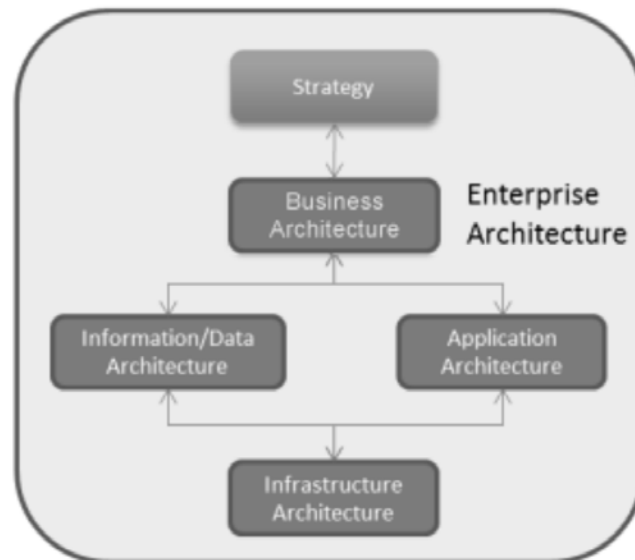


Abbildung 2.5: Traditional Enterprise Architecture Focus Areas. Quelle: [Cam14]

Des Weiteren besteht der Vorteil von Unternehmensarchitekturen darin, eine Brücke zwischen der Strategie eines Unternehmens und der tatsächlichen Umsetzung zu schaffen. Diese Umsetzung äußert sich in der Nutzung der benötigten IT-Systeme, Optimierung der Prozesse sowie in der Beschaffung der technischen Infrastruktur. Des Weiteren zeichnet sich eine Architektur dadurch aus, dass der zukünftige Zustand, der Soll-Zustand, modelliert werden kann.

Zusätzliche Vorteile einer Unternehmensarchitektur bestehen in der Möglichkeit, Geschäftsprozesse zusammenzulegen und klare Geschäftsregeln zu formulieren. Ebenso dient sie dazu, die Qualität der Unternehmensdaten zu verbessern [Cam14].

Vorteile eines Unternehmensarchitekturmanagements

Die Vorteile des Managements einer Unternehmensarchitektur basieren auf den zuvor beschriebenen Vorteilen einer Unternehmensarchitektur.

Der erste allgemeine Vorteil äußert sich in der Transparenz, die das Management der Unternehmensarchitektur schafft. Dies ist der Fall, da die Landschaft der IT-Systeme besser verstanden werden kann und Abhängigkeiten schneller deutlich werden. Hinzu kommt, dass eine Dokumentation der bestehenden Unternehmensarchitektur ermöglicht wird. Folglich dient das Enterprise Architecture Management ebenfalls dazu, Optimierungspotenziale in der Unternehmenslandschaft zu erkennen und somit zukünftige Visionen der Unternehmensarchitektur zu formulieren. Zusätzlich sind Unternehmen in der Lage, nur die IT-Systeme zu beschaffen, die einen Nutzen bringen.

Weiterhin ist es möglich, Prinzipien und Standards für eine Unternehmensarchitektur zu definieren. Aufgrund dieser beschriebenen Vorteile impliziert das Enterprise Architecture Management weitere Chancen für Unternehmen. Diese bestehen z.B. in Kostenreduktionen, reduzierten Komplexitäten in der Unternehmensarchitektur oder in der Unterstützung von Geschäftsstrategien [ASML12].

2.2.3 Unterstützung für Big Data

Heutige Projekte in Unternehmen, welche mit Big Data arbeiten und an denen mehrere Stakeholder beteiligt sind, benötigen eine gemeinsame Praxis zur erfolgreichen Gestaltung von Projekten. Diese unternehmensweite Projektsteuerung, welche auf die Geschäftsziele ausgerichtet ist, wird unter dem Begriff Enterprise Project Governance zusammengefasst [DR12]. Der Vorteil darin besteht, dass nicht nur die üblichen Faktoren bei der Projektplanung (Zeit, Kosten, Qualität) berücksichtigt werden, sondern noch weitere Faktoren. Hierbei handelt es sich zum Beispiel um Auswirkungen für das Unternehmen und die Kunden, die Zufriedenheit der Stakeholder sowie zukünftige Geschäftsmöglichkeiten [AAKE13]. Wie bereits im Abschnitt über die Vorteile einer Unternehmensarchitektur (Kapitel 2.2.2) beschrieben, dient eine Unternehmensarchitektur als Brücke zwischen der

Formulierung einer Strategie sowie deren Umsetzung [Cam14]. Dies wird in Abbildung 2.6 visualisiert.

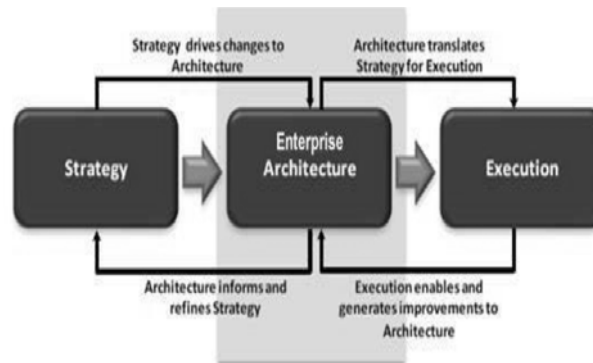


Abbildung 2.6: Enterprise Architecture as The Bridge between Strategy and Execution.
Quelle: [Cam14]

Des Weiteren kann eine Unternehmensarchitektur dabei helfen, Big-Data-Projekten ein klares Geschäftsziel zuzuweisen. Dies ist der Fall, da eine Unternehmensarchitektur ein Projektportfoliomanagement beinhaltet, welches Projekten Prioritäten und Ressourcen zuweisen kann [Cam14].

Hinzu kommt, dass eine Unternehmensarchitektur zur Planung des Umfangs von Big-Data-Projekten dient. Dies ist der Fall, da sie eine zukünftige Vision darstellt, welche von der IT sowie allen anderen Stakeholdern geteilt wird. Folglich gestaltet sich die regelmäßige Abstimmung und Koordination unter allen Projektbeteiligten effizienter.

Zusätzlich dient eine Unternehmensarchitektur dazu, die bereits im Kapitel 2.1.6 beschriebenen Informationssilos zu vereinheitlichen. Somit ist es möglich, einen unternehmensweiten Datenzugriff zu gewährleisten, bei dem agil der Datenbestand erweitert werden kann [Sch04].

Ebenfalls stellt die Unternehmensarchitektur ein Werkzeug zur Auswahl der richtigen Technologien und Tools dar. Hierbei ist es möglich, den Ist-Zustand der Architektur mit neuen Big-Data-Technologien zu vergleichen und festzustellen, was für einen Geschäftswert diese bieten. Hier dient die Unternehmensarchitektur zusätzlich dazu, Mängel bezüglich des Personals zu identifizieren. Folglich kann das nötige Personal beschafft werden [Cam14].

2.3 Modellierung

ArchiMate

ArchiMate stellt eine Modellierungssprache für Unternehmensarchitekturen dar, welche am Telematica Instituut in Enschede entwickelt und später von der Open Group standardisiert wurde. ArchiMate verfügt über eine Vielzahl von Möglichkeiten zur Darstellung einzelner Geschäftsbereiche sowie deren Beziehungen untereinander. Hierbei können beispielsweise Geschäftsprozesse, Informationsflüsse sowie IT-Systeme modelliert werden. ArchiMate dient somit als Grundlage zur Bewertung von einzelnen Geschäftsbereichen des Unternehmens [Grob]. Ein Modell, basierend auf ArchiMate, setzt sich aus verschiedenen Bestandteilen zusammen (Aspekte, Ebenen, Elemente, Beziehungen), die im weiteren Verlauf der Arbeit genauer vorgestellt werden.

Schichten

Der Kern der Modellierungssprache ArchiMate besteht aus verschiedenen Bestandteilen, welche innerhalb einzelner Schichten genutzt werden können. Hierbei handelt es sich um drei Schichten/Ebenen, die den Kern von ArchiMate zusammenfassen:

1. Der Business Layer ist die erste Schicht. Diese Schicht repräsentiert Elemente, welche im Zusammenhang mit Geschäftsprozessen und ihren ausführenden Akteuren (Personen) stehen.
2. Der Application Layer stellt eine Unterstützung des Business Layers dar, indem Anwendungen (Applikationen, Software) modelliert werden können.
3. Der Technology Layer visualisiert technologische Elemente. Dies sind Komponenten wie Betriebssysteme, Hardware, Netzwerke oder Speichersysteme.

Des Weiteren existieren weitere Ebenen, die durch das ArchiMate Full Framework ergänzt werden. Dieses wird in Abbildung 2.7 dargestellt.

Die erste Schicht, die durch ArchiMate Full ergänzt wird, ist der Physical Layer. Dieser ist eine Erweiterung der Technologieebene, welcher zusätzlich physikalische Komponenten (Equipment, Material, verteilte Netzwerke, Gebäude/Fabriken) modelliert. Der Motivational Layer modelliert Motivationen oder Gründe für Entwurfsentscheidungen oder Änderungen an der Unternehmensarchitektur (Ziele, Anforderungen, Bedingungen,

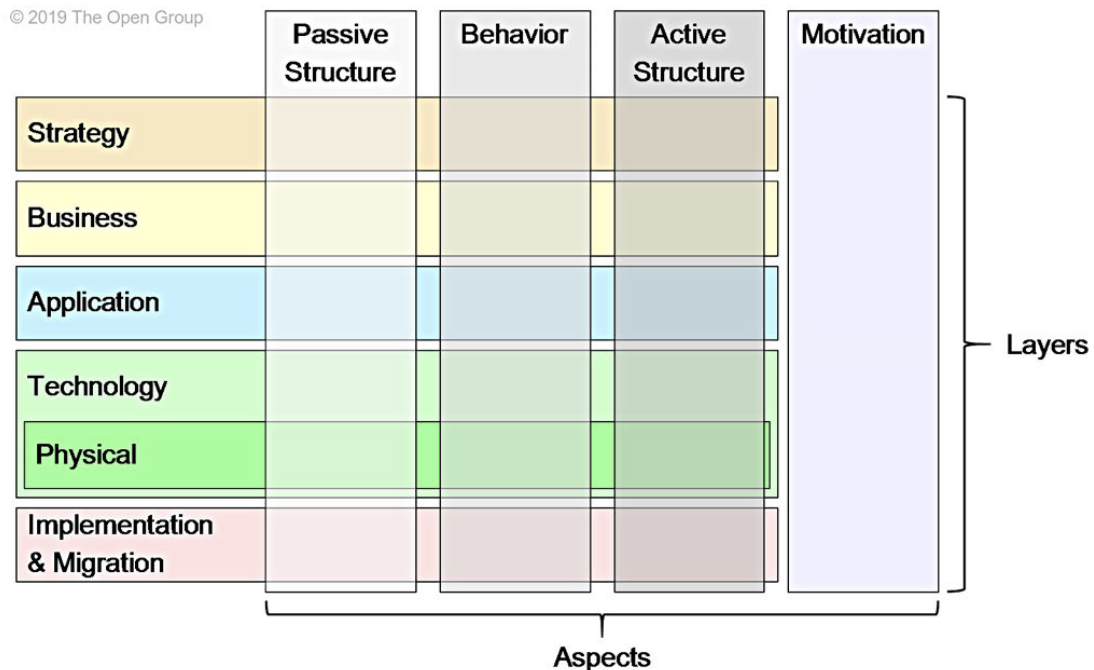


Abbildung 2.7: ArchiMate Full Framework. Quelle: [Groa]

Prinzipien, Werte etc.). Der Strategy Layer dient dazu, Auswirkungen auf die Unternehmensarchitektur hinsichtlich der strategischen Ausrichtung eines Unternehmens darzustellen und welche Ressourcen/Fähigkeiten dafür nötig sind. Der Implementation und Migration Layer dient vor allem dazu, Implementationen von Unternehmensarchitekturen sowie den Vorgang zur Veränderung dieser zu visualisieren. Ermöglicht wird dies durch Komponenten wie Arbeitspakete oder Plateaus.

Aspekte

ArchiMate ermöglicht es, verschiedene Blickwinkel innerhalb eines Unternehmens einzunehmen. Dies wird neben der Unterteilung in Ebenen unter anderem dadurch ermöglicht, dass alle Ebenen in ArchiMate zusätzlich in Aspekte unterteilt werden (in Abbildung 2.7 dargestellt). Der Active Structure Aspekt stellt aktive Elemente (Akteure, Anwendungen etc.) dar. Der Behavior Aspekt verkörpert das Verhalten der Akteure mittels Komponenten wie Prozesse, Funktionen, Ereignisse etc. Der Passive Structure Aspekt repräsentiert Objekte, an denen das Verhalten ausgeführt wird (Datenobjekte, Dokumente etc.) [Groa].

Elemente

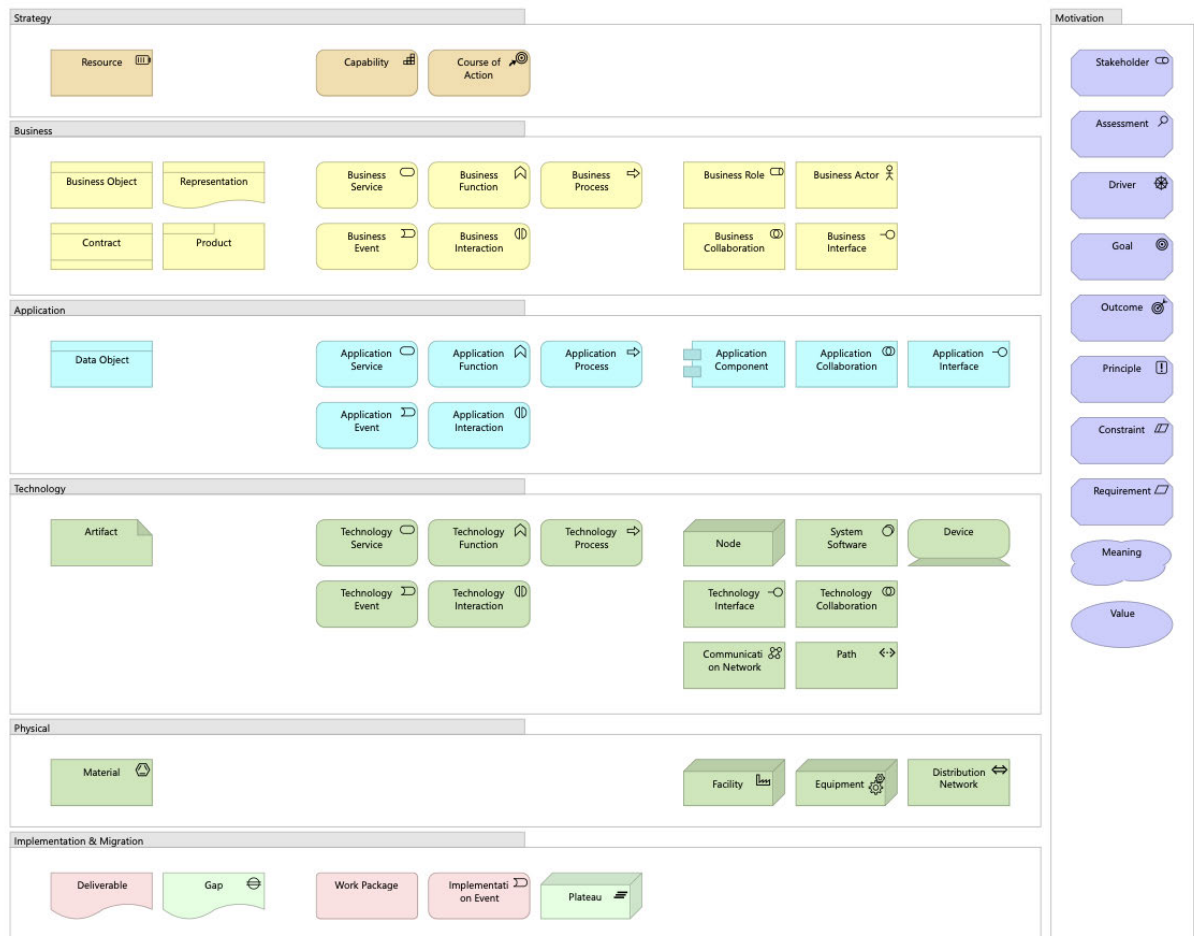


Abbildung 2.8: Elemente in ArchiMate Full. Quelle: [Ram19]

Die Abbildung 2.8 visualisiert alle Elemente, welche mittels des ArchiMate Full Frameworks darstellbar sind. Links befinden sich hierbei die Active Structure Elements, mittig die Behavior Elements und rechts die Passive Structure Elements.

Beziehungen

ArchiMate definiert einen Satz an Beziehungen, welcher sich klassifizieren lässt:

1. Strukturelle Beziehungen, welche statische Zusammenhänge modellieren.
2. Abhängigkeitsbeziehungen, welche die gegenseitige Unterstützung zwischen Elementen visualisieren.
3. Dynamische Beziehungen, die dazu dienen, zeitliche Abhängigkeiten zwischen Elementen darzustellen.
4. Zusätzliche Beziehungen, die nicht genauer definiert sind. Hierzu gehören die Vererbung sowie die Assoziation [Groa].

Einen Überblick dieser Beziehungen sowie deren Konnektoren liefert die Abbildung 2.9.

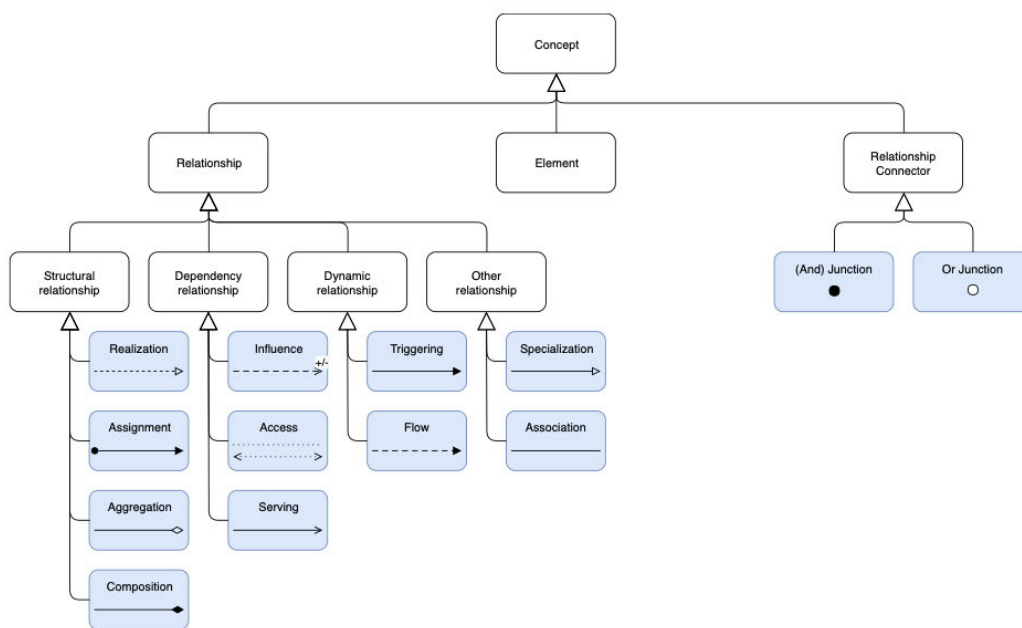


Abbildung 2.9: Beziehungen in ArchiMate. Quelle: [Ram19]

2.4 Metamodellierung

Bei der Metamodellierung handelt es sich um ein Verfahren, bei dem vorhandene Modelle selbst zum Gegenstand der Modellierung werden [AK03]. Dieser Vorgang wird in Abbildung 2.10 dargestellt.

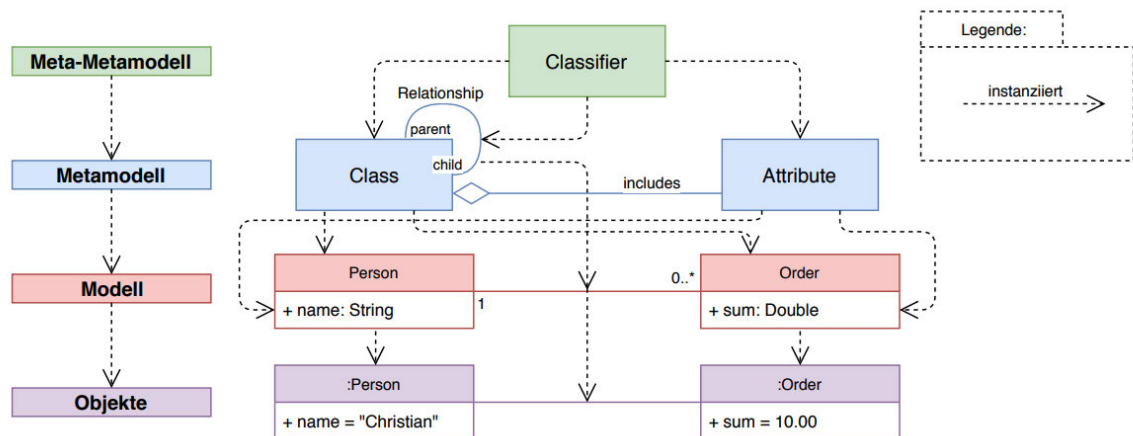


Abbildung 2.10: Metamodellierung an einer exemplarischen Modellsprache. Quelle: [Ram19]

Das Prinzip der Metamodellierung ist hierbei am Beispiel einer exemplarischen Modellierungssprache zu erkennen. Dabei ist das Konzept der Abstraktion durch mehrere Ebenen zu erkennen.

2.4.1 Metamodell in ArchiMate

Die Modellierungssprache ArchiMate basiert ebenfalls auf einem Metamodell. Dieses dient dazu, die einzelnen Elementtypen zu abstrahieren und deren erlaubte Verknüpfungen untereinander zu visualisieren. Dieses generische Metamodell ist in Abbildung 2.11 dargestellt.

Die einzelnen Elemente sowie Beziehungen, welche in Abbildung 2.11 dargestellt sind, stellen eine Abstraktion dar und repräsentieren das generische Metamodell. Folglich dürfen Instanzen dieses Metamodells nur jene Arten von Beziehungen und Elementen verwenden, welche in diesem Metamodell festgelegt sind. Jede einzelne Schicht (Business Layer, Application Layer, Technology Layer) besitzt zusätzlich ein eigenes Metamodell,

© 2019 The Open Group

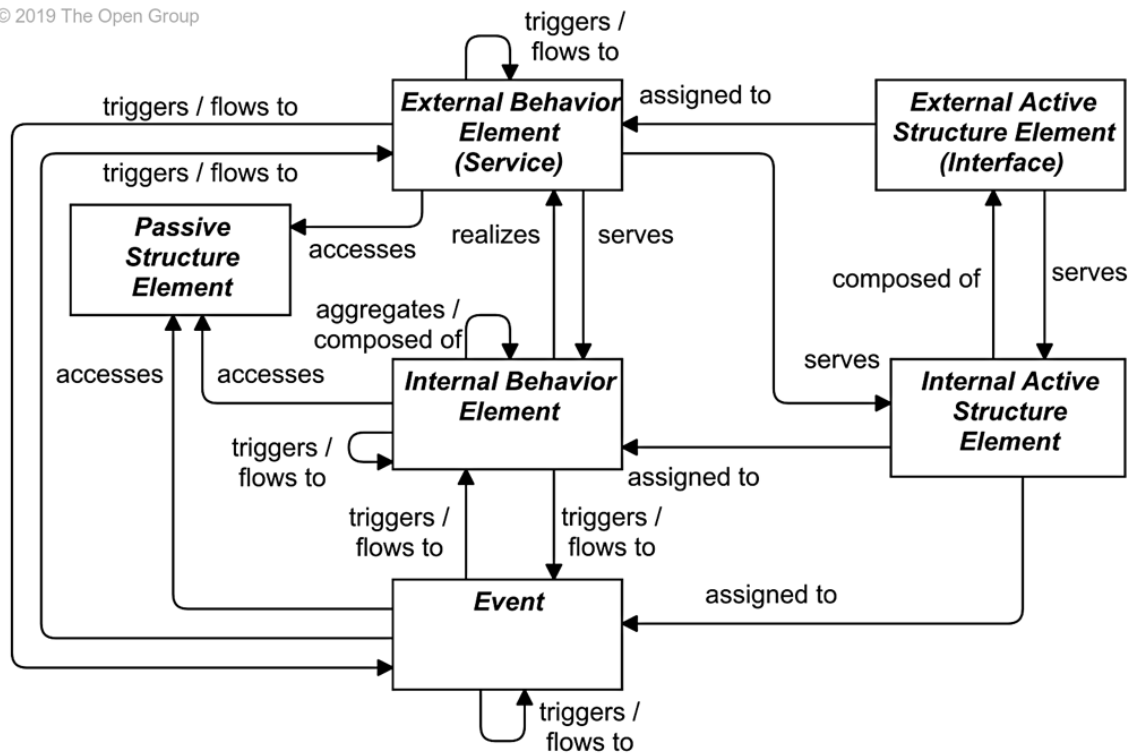


Abbildung 2.11: Das Metamodell der Verhaltens- und Strukturelemente in ArchiMate.
Quelle: [Groa]

welches analog zu dem generischen Metamodell erstellt wird. Die einzelnen Metamodelle in Kombinationen mit den Beziehungen untereinander ergeben das komplette ArchiMate-Metamodell [Groa].

3 Anforderungsanalyse

Dieses Kapitel liefert einen Überblick über die Anforderungen an das Metamodell bezüglich des Umgangs mit Big Data in Unternehmen, um folglich das Metamodell in ArchiMate zu bewerten und gegebenenfalls anpassen zu können. Dabei werden zunächst allgemeine Anforderungen aufgelistet. Diese Anforderungen basieren dabei auf den Erkenntnissen der vorherigen Kapitel und sollen in konkreten Modellen realisiert werden können.

3.1 Allgemeine Anforderungen

Bei den allgemeinen Anforderungen handelt es sich um grundlegende Anforderungen, welche bei der Bewertung und Anpassung des Metamodells zu berücksichtigen sind. Diese Anforderungen dienen unter anderem dazu, Verständnis für den Modellierer zu schaffen, die Anpassbarkeit des Modells zu ermöglichen und den Nutzen während einer Modellierung zu erhöhen:

1.) Das Modell soll auf dem Metamodell und dem Standard von ArchiMate 3.1 basieren

Der Modellierungsstandard und die damit einhergehende Syntax und Semantik von ArchiMate 3.1 stellen die Grundlage für die Entwicklung des Metamodells dar. Das Gleiche gilt für das Metamodell sowie für die Bestandteile der ArchiMate Modellierungssprache (z.B. Ebenen, Elemente, Beziehungen etc.)

2.) Erweiterungen sollen vom Metamodell realisiert werden

Das Modell soll notwendige Ergänzungen (Entitäten, Relationen, Layer etc.), die für die Abbildung datenintensiver Prozesse nötig sind, beinhalten.

3.) Mit dem entwickelten Metamodell können weiterhin ArchiMate Modelle erstellt werden

Basierend auf Anforderung 1, soll es dem Nutzer möglich sein, weiterhin Modelle in ArchiMate zu modellieren. Das entwickelte Metamodell stellt somit eine Erweiterung dar, welche auf den Nutzen von ArchiMate aufbaut und notwendige Modifikationen für die Abbildung von datenintensiven Prozessen beinhaltet.

4.) Das Metamodell soll eine Wiederverwendung gewährleisten

Die Wiederverwendung stellt hierbei ein ökonomisches Ziel dar, da Fehler in der Modellierung vermieden werden können. Dies soll durch Abstraktion geschehen. Umgesetzt werden soll diese Abstraktion in der konkreten Modellierung durch zwei Beziehungen, der Spezialisierung und Realisierung von Elementen.

5.) Die Darstellung eines Ist-Zustandes ist gewährleistet

Mittels des Metamodells soll der Ist-Zustand der einzelnen Komponenten sowie deren Beziehungen zueinander deutlich werden. Folglich soll erkennbar sein, welche Elemente/Entitäten usw. existieren und in welcher Relation diese zueinander stehen. Dies dient als Grundlage dafür, mögliche Veränderungen für einen zukünftigen Soll-Zustand abzuleiten.

3.2 Individuelle Anforderungen

Basierend auf den allgemeinen Anforderungen sowie den Problemen und Herausforderungen mit Big Data, werden in diesem Kapitel folgende Anforderungen für das Metamodell definiert:

1.) Big Data Spezialisten sollen eine übergeordnete Rolle in der Unternehmensarchitektur spielen

Berufe wie ein Data Scientist, Big Data Architect etc. sollen mittels einer eigenen Entität eine primäre Rolle im Metamodell spielen. Des Weiteren sollen innerhalb von Big-Data-Projekten Experten aus einzelnen Geschäftsbereichen vorhanden sein. Zusätzlich sollen Data Scientists sowie Analysts am datenintensiven Prozess beteiligt sein.

2.) Datenlastige Prozesse sollen ein Bestandteil des Metamodells sein

In jedem Big-Data-Projekt müssen datenlastige Prozesse ein fester Bestandteil sein. Hierzu gehören Prozesse der Datenerfassung, Datenspeicherung, Datenvorbereitung und Datenverarbeitung. Dies soll im Modell repräsentiert werden, um die Datenqualität gewährleisten zu können.

3.) Big-Data-Konzepte müssen in Projekten verstanden werden

Um die Arbeit in Big-Data-Projekten erfolgreich gestalten zu können, müssen die Konzepte von Big Data verstanden, genutzt und geschult werden. Hierbei soll das Metamodell helfen, indem die notwendigen Bestandteile modelliert werden können.

4.) Das Modell soll eine Modellierung von Echtzeitaspekten ermöglichen

Wie bereits im Kapitel zu den Problemen und Herausforderungen von Big Data beschrieben, stehen heutige Unternehmen vor der Aufgabe, Daten in Echtzeit zu verarbeiten und Analysen mit diesen Daten in Echtzeit zu tätigen. Dies soll dadurch realisiert werden, indem Technologien und Konzepte modelliert werden können, welche die Datenverarbeitung in Echtzeit ermöglichen. Das Metamodell soll hierfür die notwendigen Modellierungselemente zur Verfügung stellen.

5.) Das Metamodell soll eine Big-Data-Plattform darstellen können

Um der Arbeit mit datenintensiven Prozessen gerecht zu werden, wird eine Big Data Plattform benötigt, welche die notwendigen Applikationen sowie Tools zur Verfügung stellt. Hierzu gehören beispielsweise Analysetools, Tools zur Datenvisualisierung oder Applikationen zur Datenextraktion. Diese gemeinsam nutzbare Analyseplattform soll dazu dienen, dass die bereits beschriebenen Informationssilos in Unternehmen umgangen werden.

6.) Das Metamodell soll Applikationen zur Datenselektion abbilden

Diese soll primär dazu dienen, nur projektrelevante Daten auszuwählen werden, damit diese weiter verarbeitet werden können.

7.) Self-Learning-Tools sollen modelliert werden

Diese Anforderung korrespondiert mit Anforderung 3 der individuellen Anforderungen. Das Metamodell soll folglich die notwendigen Tools und Applikationen abbilden können.

8.) Das Metamodell soll ein Big-Data-Ökosystem abbilden können

Wie in Kapitel 2.1.6 dargestellt, gehören zu einem Big-Data-Ökosystem die richtige Hardware sowie die richtigen Frameworks. Diese sind notwendig, um mit den Datenmengen arbeiten zu können. Das Metamodell soll diese Komponenten abbilden.

9.) Verschiedene Systeme zur Speicherung und Extraktion sollen modelliert werden können

Um die Datenmengen verarbeiten zu können, bedarf es verschiedener Speichersysteme sowie Datenquellen. Hierbei soll es sich um die in Kapitel 2.1.5 vorgestellten Technologien handeln. Hierzu gehören beispielsweise NoSQL-Datenbanken, In-memory Computing, Cloud Computing oder Apache Hadoop. Das Metamodell soll einen Komponententyp enthalten, welcher diese Systeme repräsentieren kann.

10.) Technologien müssen individuell für ein Big-Data-Projekt anpassbar sein

Aufgrund verschiedener Anforderungen und Ziele einzelner Big-Data-Projekte, sind individuelle Technologien nötig. Folglich müssen je nach Projektziel und Projektumfang verschiedene Technologien verwendet werden. Hierbei soll das Metamodell helfen, indem Elemente für diese Technologien modelliert werden können. Demnach können individuelle Big-Data-Technologien für unterschiedliche Projekte in einem konkreten Modell instanziiert werden.

11.) Das Metamodell soll eine verteilte Datenverarbeitung ermöglichen

Hierbei soll eine Replikation der Hardware sowie die Erweiterbarkeit der Systeme gewährleistet werden. In der konkreten Instanz soll dies technisch beispielsweise durch Cloud Computing und paralleler Programmierung mit MapReduce realisierbar sein.

12.) Das Modell soll dazu dienen, Schwachstellen bezüglich der Datensicherheit aufzuzeigen

Wie bereits in Kapitel 2.1.6 dargestellt, stellt die Sicherheit von personenbezogenen Daten oftmals ein Problem in Projekten dar. Mögliche konzeptionelle Ansätze des Metamodells sollen dazu dienen, solche Probleme zu erkennen und auszuräumen.

13.) Das Modell soll die Datenkommunikation abbilden

Das Metamodell soll hierbei eine Möglichkeit bieten, die Datenkommunikation zwischen Systemen sowie Personen in datenintensiven Prozessen abbilden zu können.

14.) Die Datendiversität soll visualisiert werden können

Wie bereits im Kapitel 2.1.2 erklärt, ist die Datenvielfalt ein Merkmal von Big Data. Diese Vielfalt der Daten soll mittels des Metamodells abbildbar sein können. Dies soll durch Datenelemente realisiert werden, welche einzelne Datentypen repräsentieren.

15.) Die Lokalität der Daten soll abgebildet werden können

Im Kapitel zu den Technologien wird deutlich, dass Daten an verschiedenen Orten gespeichert und extrahiert werden können. Das Metamodell soll diese Orte modellieren können.

4 Konzeption

Aufbauend auf Kapitel 3, werden im weiteren Verlauf der Arbeit mögliche Erweiterungen des Metamodells in ArchiMate ermittelt. Hierbei wird sich auf die individuellen Anforderungen (siehe Kapitel 3.2) an das Metamodell fokussiert. Dies ist der Fall, da diese voraussichtlich eine Erweiterung des bestehenden Metamodells implizieren. Folglich handelt es sich bei den allgemeinen Anforderungen um Anforderungen, welche durch das bestehende Metamodell in ArchiMate realisiert werden. Als Grundlage für die Erweiterung des Metamodells werden Viewpoints verwendet. Ein Viewpoint stellt in ArchiMate einen Auszug einer Teilmenge dar, welche intern Elemente und deren Beziehungen untereinander darstellt [Groa]. Die Viewpoints werden, falls es nötig ist, angepasst.

Des Weiteren werden zwei Views entworfen. Views realisieren in ArchiMate einen Ansatz, bei dem Stakeholder die Unternehmensarchitektur aus ihrer eigenen Sicht betrachten können [Groa]. Views dienen somit dazu, die Anliegen der Stakeholder zu behandeln. Dabei werden Views von den Viewpoints gesteuert [ISO11]. Die Konzepte hinter den Views und Viewpoints sowie deren Beziehungen untereinander werden in Abbildung 4.1 visualisiert. Basierend auf diesen Entwürfen und Konzeptionen, wird das bestehende Metamodell in Archimate bewertet und gegebenenfalls erweitert.

In diesem Kapitel werden bestimmte Probleme und Herausforderungen aus Kapitel 2.1.6 ausgewählt, um diese mit den ausgewählten Viewpoints und den Views zu behandeln.

Abbildung 4.2 visualisiert den Business Process Cooperation Viewpoint in ArchiMate. Dieser wird dafür genutzt, um Modifikationen vorzunehmen, welche in Abbildung 4.3 dargestellt sind. In Kapitel 3 besteht eine Anforderung an das Metamodell darin, datenintensive Prozesse im Modell zu integrieren. Dies hat den Vorteil, die Datenqualität in Projekten durch Prozesse zur Datenvorverarbeitung gewährleisten zu können. Die Idee hierbei ist, dass Geschäftsprozesse die zuvor beschriebenen datenintensiven Prozesse nutzen können. Der Data Scientist soll hierbei ein fester Bestandteil dieser Prozesse

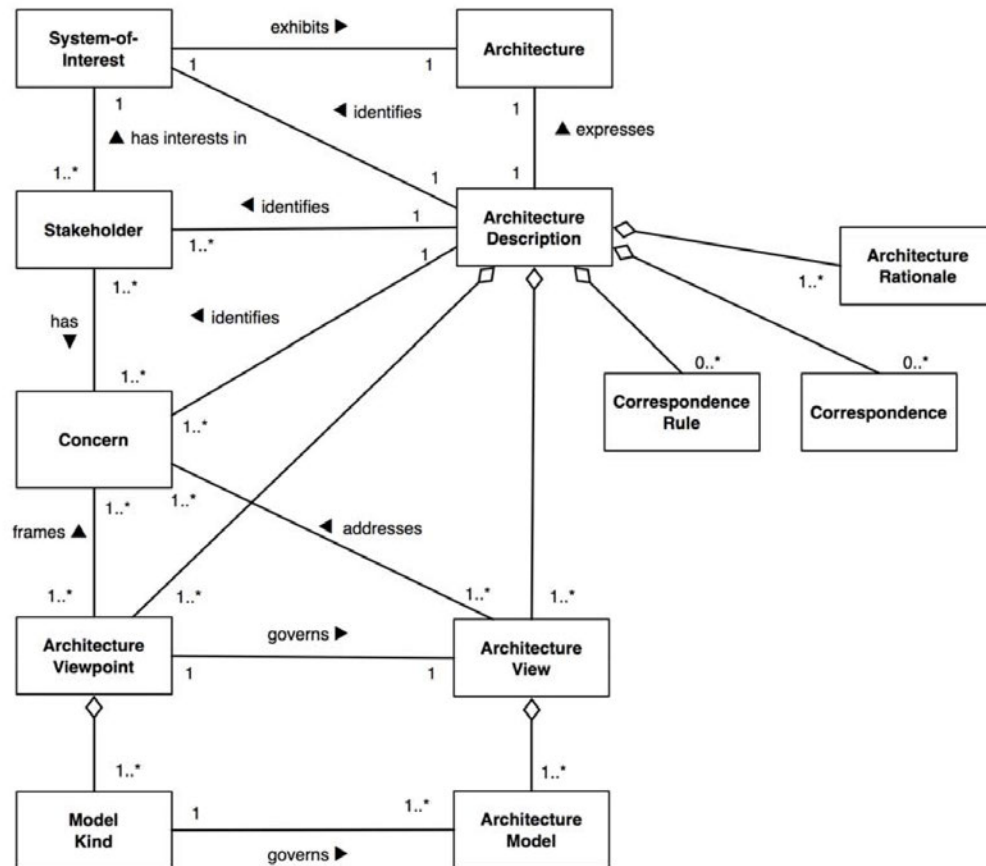


Abbildung 4.1: The meta model of concepts provided by the ISO/IEC/IEEE 42010 Standard. Quelle: [ISO11]

sein. Zusätzlich ist es notwendig, Experten aus einzelnen Geschäftsbereichen zu integrieren. Abbildung 4.3 visualisiert diese Anpassungen. Die vorgenommenen Anpassungen des Viewpoints lassen sich mittels des bestehenden Metamodells in ArchiMate und deren elementaren Bestandteilen darstellen. Folglich bedarf es hierbei keiner weiteren Erweiterungen im Metamodell. Der in Grau dargestellte datenintensive Prozess lässt sich ebenfalls mit den sprachlichen Elementen in ArchiMate modellieren.

Ein weiterer Ansatz zur Modifikation bietet der Application Usage Viewpoint, welcher in Abbildung 4.4 zu sehen ist.

Weitere Anforderungen aus Kapitel 3 sollen über Anpassungen dieses Viewpoints umgesetzt werden. Diese sind in Abbildung 4.5 realisiert. Die Anforderungen sollen über

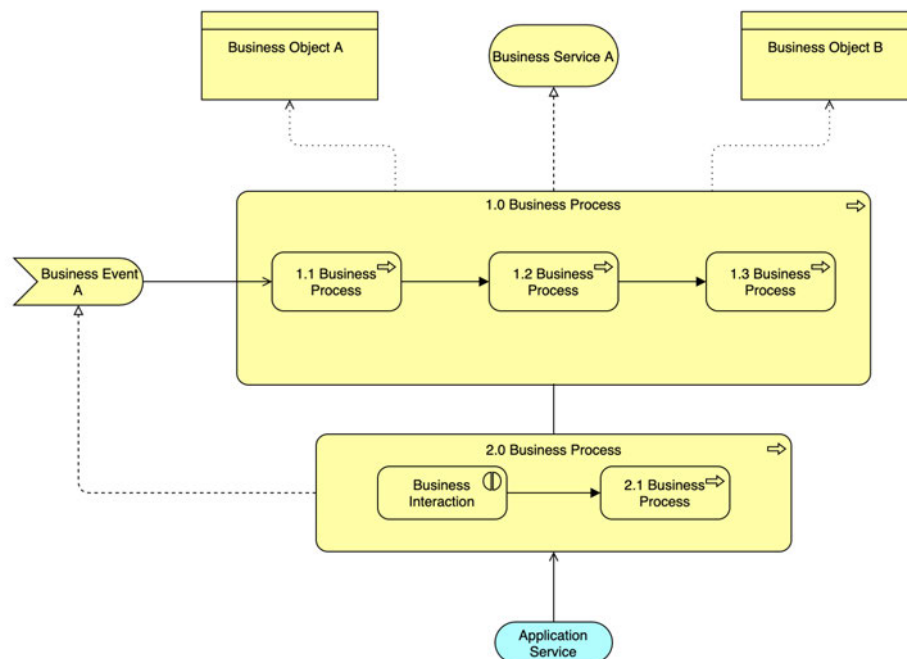


Abbildung 4.2: Business Process Cooperation Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]

eine so genannte Big-Data-Plattform realisiert werden. Eine der Anforderungen besteht darin, den Umgang mit Big-Data-Konzepten zu erlernen und zu verstehen. Dies soll über eine Self-Learning-Applikation der Plattform geschehen. Des Weiteren soll die Plattform die Datenvisualisierung ermöglichen, welche mittels Visualisierungsapplikationen realisiert wird. Zusätzlich soll die Plattform dazu dienen, aus den großen Datenmengen die notwendigen Daten für ein Projekt zu extrahieren, zu selektieren und die Daten somit vorzubereiten. Folglich können z.B. Inkonsistenzen der Daten vermieden werden. Hierbei werden ebenfalls nur projektrelevante Daten verwendet. Dies wird über Applikationen zur Datenvorbereitung umgesetzt. Damit diese Plattform in datenintensiven Prozessen genutzt werden kann, realisiert die Plattform einen Service. Diese vorgenommenen Anpassungen lassen sich ebenfalls durch das bereits existierende Metamodell visualisieren. Bei der Plattform handelt es sich lediglich um verschiedene Applikationen, welche durch die Elemente in ArchiMate darstellbar sind. Der Plattform Service ist hierbei ebenfalls

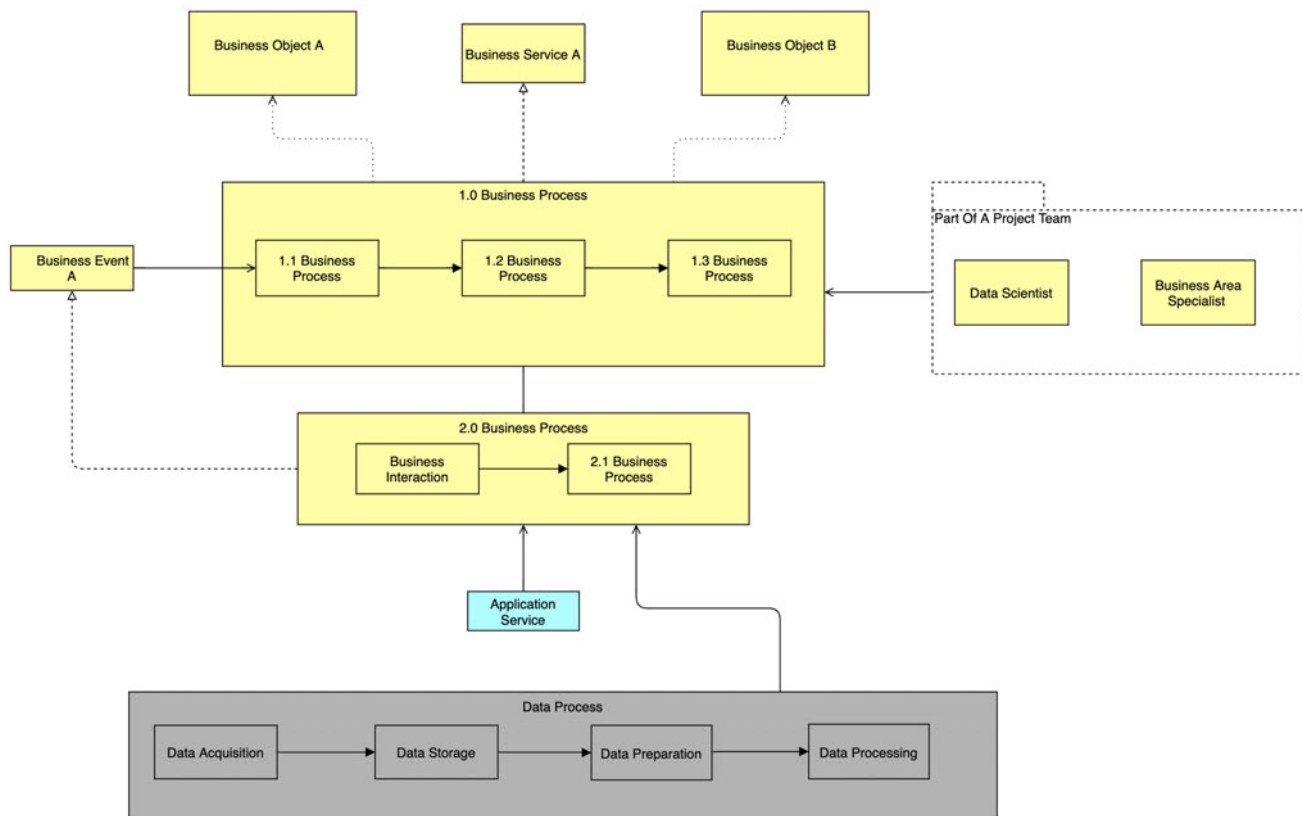


Abbildung 4.3: Modifizierter Business Process Cooperation Viewpoint in ArchiMate.
Quelle: Tom Krause (2020)

ein Applikationsservice. Folglich sind auf Basis dieses erweiterten Viewpoints keine Erweiterungen des Metamodells nötig.

Zusätzlich besteht die Schwierigkeit in Big-Data-Projekten, Daten aus verschiedenen Quellen in Echtzeit zu verarbeiten und in Echtzeit Aussagen über diese Daten zu treffen. In Abbildung 4.6 ist der Layered Viewpoint in ArchiMate dargestellt, welcher für diese Anforderung angepasst wird.

In Abbildung 4.7 sind diese Anpassungen des Layered Viewpoints zu sehen. Die ersten Modifikationen bestehen darin, einen Data Layer einzuführen. Der Data Layer dient dazu, eine Pipeline zu integrieren, welche intern eine Echtzeitverarbeitung der Daten realisiert. Diese Pipeline nimmt die Daten aus der Quelle entgegen, um sie anschließend zu verarbeiten und anzureichern. Im letzten Schritt werden die Daten dann bereitgestellt. Hierbei greift ein Analyst, beispielsweise ein Data Analyst/Data Scientist auf diese Da-

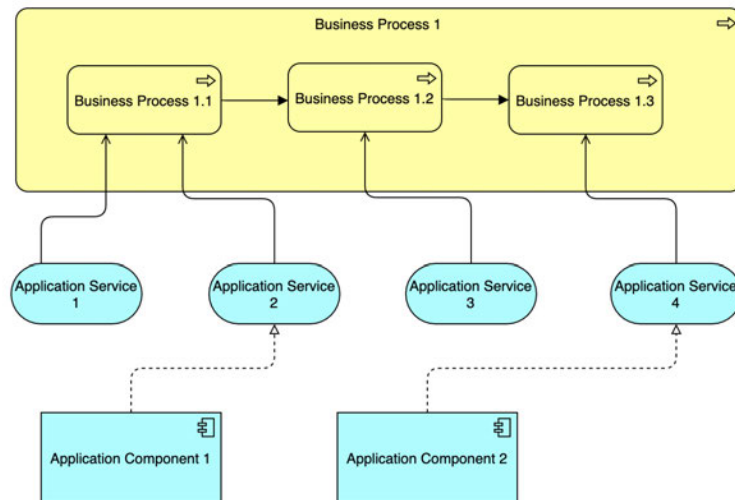


Abbildung 4.4: Application Usage Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]

ten zu. Die Daten werden dann im nächsten Schritt über einen Data Layer Service zur Verfügung gestellt. Folglich können die verarbeiteten Daten über diesen Service an eine Applikation weitergereicht werden, die diese Daten benutzt. Hierbei kann es sich zum Beispiel um eine Applikation zur Datenvisualisierung handeln. Diese Modifikationen dienen dazu, datenintensive Prozesse in Echtzeit zu integrieren. Andererseits ist es jedoch ebenso möglich, die ursprüngliche Modellierung, wie in Abbildung 4.7 visualisiert, weiterhin zu nutzen. Die vorgenommenen Änderungen dieses Viewpoints implizieren eine Erweiterung des Metamodells in ArchiMate. Das bestehende Metamodell muss um einen Layer erweitert werden, den Data Layer sowie dessen Elemente und Relationen. Die lokale Anordnung des Data Layers wird noch im weiteren Verlauf der Arbeit erläutert. Dieser neu eingeführte Layer muss dann anschließend in das bestehende Metamodell in ArchiMate integriert werden.

Die Anpassungen aus Abbildung 4.7 dienen als Grundlage und Orientierung für weitere Modifikationen der Viewpoints in ArchiMate. Folglich werden diese auch weiterhin verwendet. Um die Modifikationen aus Abbildung 4.7 technisch zu realisieren, eignet sich der Technology Usage Viewpoint (Abbildung 4.8).

Basierend auf dem Technology Usage Viewpoint, können weitere Lösungsansätze für die Abbildung datenintensiver Prozesse in ein Modell integriert werden. Hierzu werden die

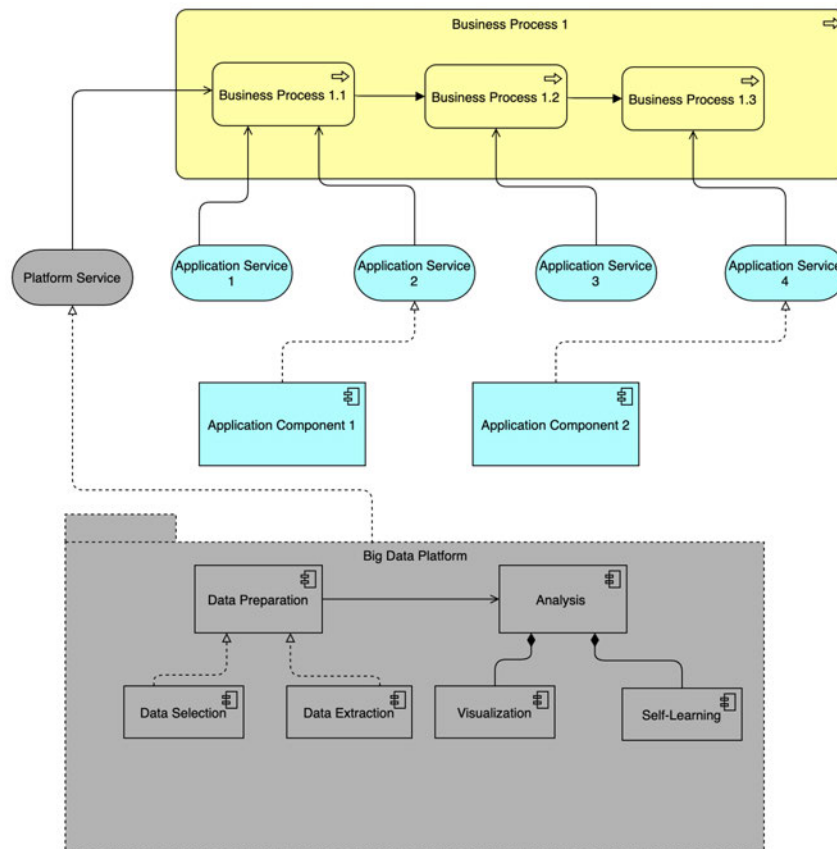


Abbildung 4.5: Modifizierter Application Usage Viewpoint in ArchiMate. Quelle: Tom Krause (2020)

technischen Herausforderungen aus Kapitel 2.1.6 behandelt. Die erste Herausforderung besteht in der Bereitstellung der technischen Infrastruktur für Big Data. Die technische Infrastruktur soll die richtige Hardware und Frameworks zur Verfügung stellen. Abbildung 4.9 stellt ein Beispiel dafür dar. Dort ist zu erkennen, dass eine Cloud im Technology Layer sowie im Data Layer integriert ist, welche die notwendige Hardware und Technologien für die Datenanalyse zur Verfügung stellt. Dazu gehören insbesondere Datenbanken, die für Big Data ausgelegt sind. Bei diesen handelt es sich beispielsweise um NoSQL-Datenbanken oder In-Memory-Datenbanken. Diese sind in der Abbildung als Big Data Databases gekennzeichnet. Des Weiteren soll die Echtzeitverarbeitung der Daten durch die Anpassungen dieses Viewpoints technisch realisiert werden. Hierzu ist eine verteilte Datenverarbeitung notwendig, die beispielsweise durch das Hadoop-Framework realisiert werden soll, welches intern mit dem MapReduce-Konzept arbeitet. Bei diesem Konzept ist

die Datenverarbeitung in Echtzeit gewährleistet. Dieser Ansatz zur Datentransformation ist bewusst in den Data Layer integriert, da dieser den Vorgang der Datenverarbeitung abbilden soll. Der Data Layer sowie der Technology Layer realisieren das in Kapitel 2.1.6 beschriebene Big-Data-Ökosystem. Ein Analyst, welcher als Element dem Data Layer zuzuordnen ist, kann die konsolidierten Daten über den zur Verfügung gestellten Data Layer Service analysieren. Des Weiteren können die verarbeiteten Daten über den Service zur weiteren Auswertung an Applikationen gereicht werden. Die Erweiterungen in Abbildung 4.9 lassen sich zum Teil durch das Metamodell in ArchiMate realisieren. Dies ist z.B. einerseits der Fall, da das Element im Data Layer eine Technologie darstellt, wofür bereits Elemente in ArchiMate existieren. Bei dem Data Layer Service wiederum handelt es sich um einen Service, der dem neu eingeführten Data Layer zuzordnen ist. Folglich ist das Metamodell aufgrund dieser Modifikationen um einen Data Layer und die zugehörigen Beziehungen zu erweitern.

Eine weitere Anforderung an das Metamodell besteht darin, die Datendiversität und die Datenkommunikation in datenintensiven Prozessen abbilden zu können. Dies soll durch die Abbildung 4.10 repräsentiert werden. Hierbei verkörpert das Data Element im Data Layer die verschiedenen Datentypen. Dieses Data Element besteht aus dem Data Object des Application Layers. Die einzelnen Datentypen sind dem Data Layer zuzuordnen. Des Weiteren findet die Datenkommunikation im Data Layer statt. Dadurch soll es möglich sein, dass die Daten in datenintensiven Prozessen zwischen Systemen (z.B. Nachbarsysteme oder Applikationen) sowie Personen kommuniziert werden können. Hierbei bedarf es wieder der Modellierung eines Data Layers und somit einer Erweiterung des Metamodells.

Ein zusätzliches Problem in datenintensiven Prozessen besteht in der Datensicherheit. Diese Datensicherheit bezieht sich insbesondere auf Personendaten. Ein Beispiel hierfür wäre ein Medienbruch. Bei diesem werden Daten, welche ursprünglich aus einer Datenquelle stammen, durch Personen ausgedruckt. Dies soll Abbildung 4.11 darstellen. Das Metamodell soll folglich in der Lage sein, solche Schwachstellen bezüglich der Datensicherheit abbilden zu können. Um diese Schwachstellen der Datensicherheit modellieren zu können, wird ebenfalls ein Data Layer benötigt. Hierbei müssen die Beziehungen zwischen den Elementen des Data Layers und des Business Layers/Application Layers realisierbar sein.

Die Anforderungen aus Kapitel 3, welche mittels des bestehenden Metamodells in ArchiMate nicht umgesetzt werden können, implizieren eine Erweiterung des Metamodells.

Es handelt sich hierbei um die Anforderungen 4, 12, 13, 14 und 15 aus den individuellen Anforderungen (siehe Kapitel 3.2). Folglich wird das Metamodell im weiteren Verlauf der Arbeit angepasst und erweitert.

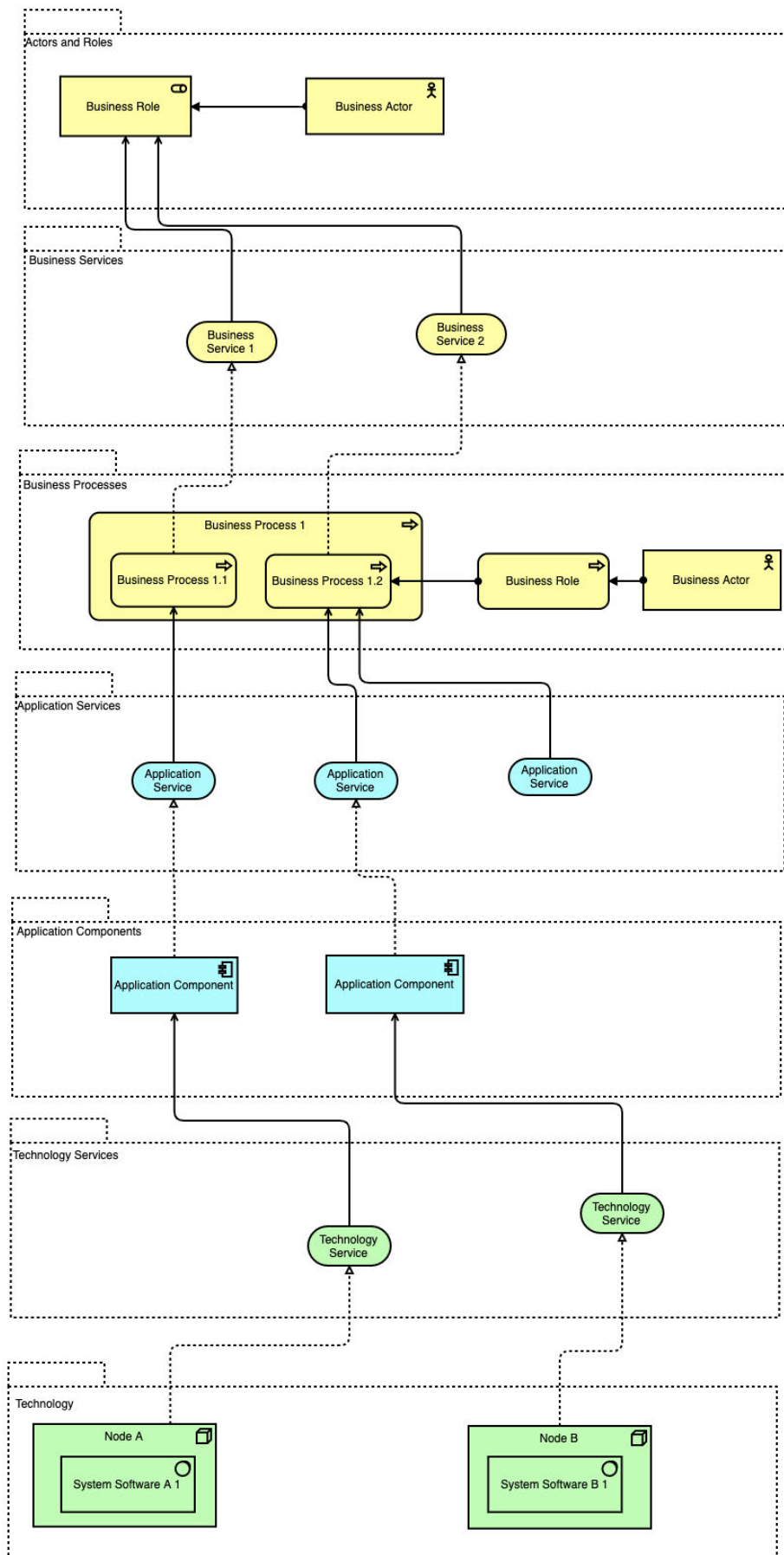


Abbildung 4.6: Layered Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]

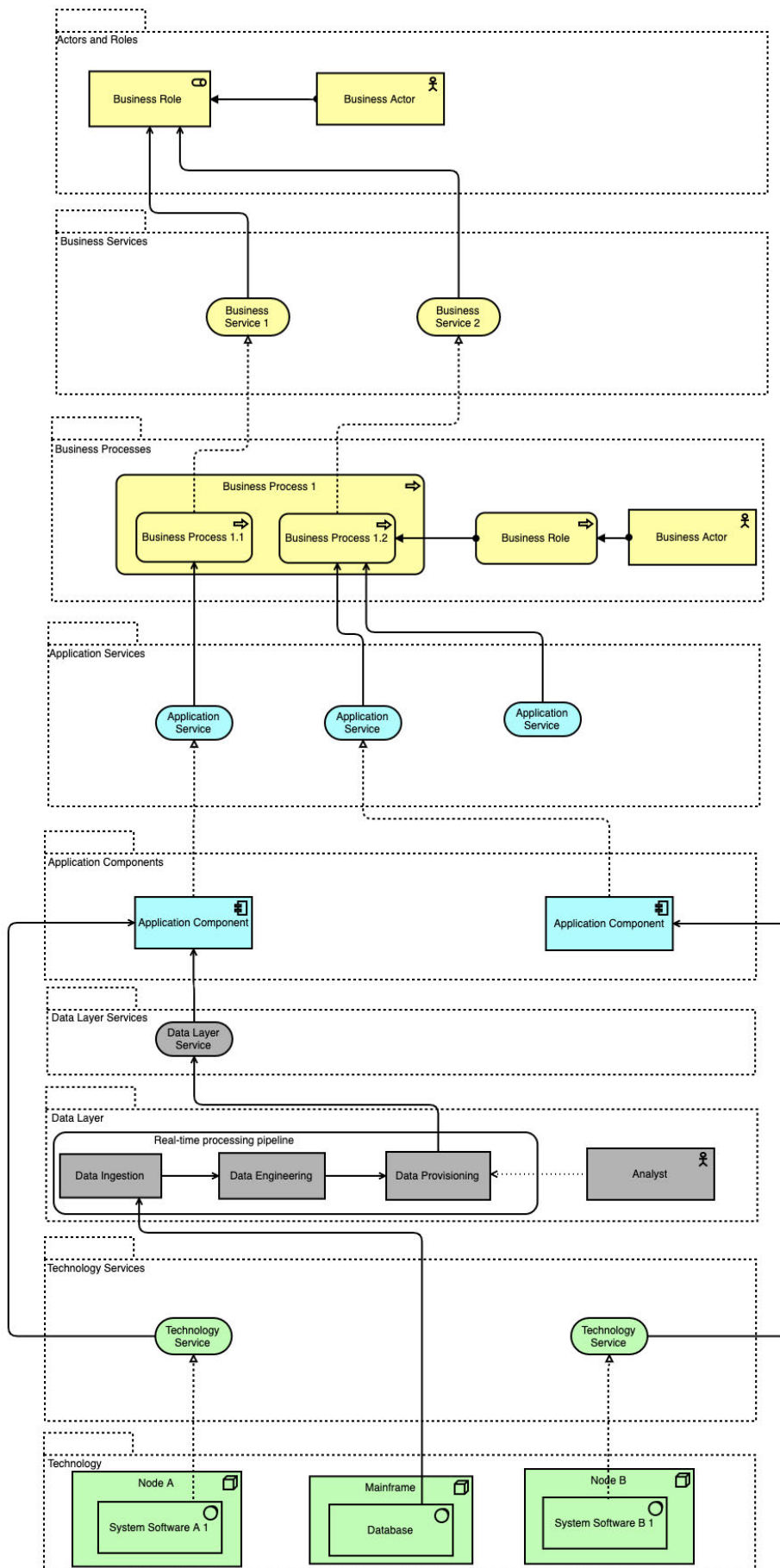


Abbildung 4.7: Modifizierter Layered Viewpoint in ArchiMate. Quelle: Tom Krause (2020)

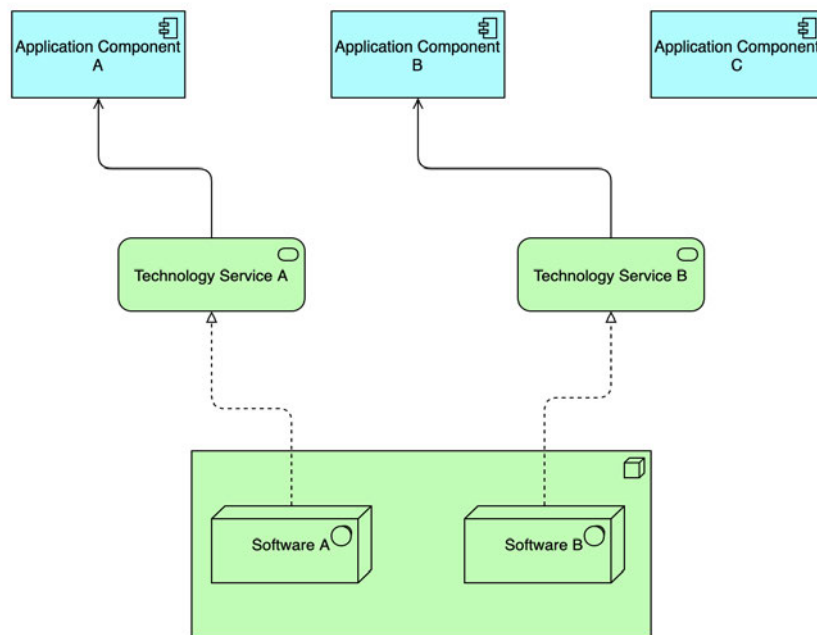


Abbildung 4.8: Technology Usage Viewpoint in ArchiMate. Quelle: In Anlehnung an [Groa]

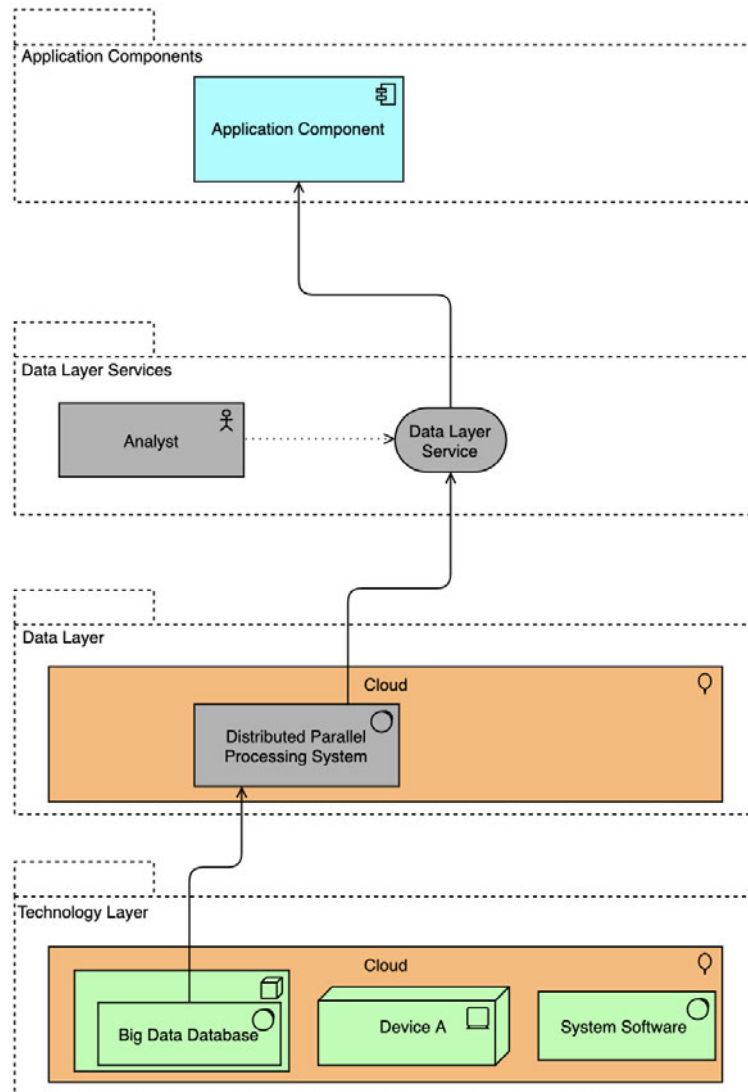


Abbildung 4.9: Modifizierter Technology Usage Viewpoint in ArchiMate. Quelle: Tom Krause (2020)

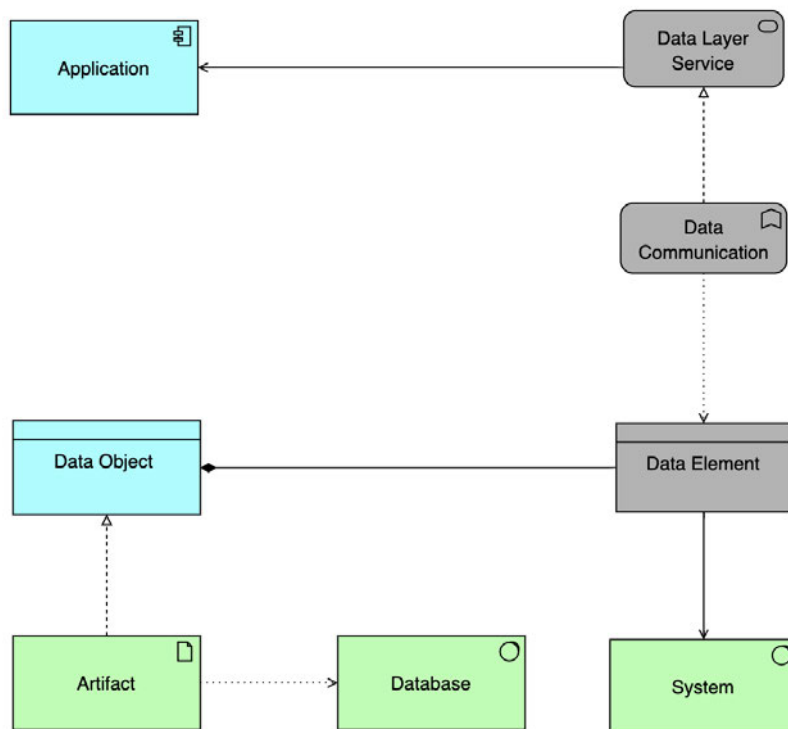


Abbildung 4.10: Datenkommunikation und Datendiversität durch den Data Layer. Quelle: Tom Krause (2020)

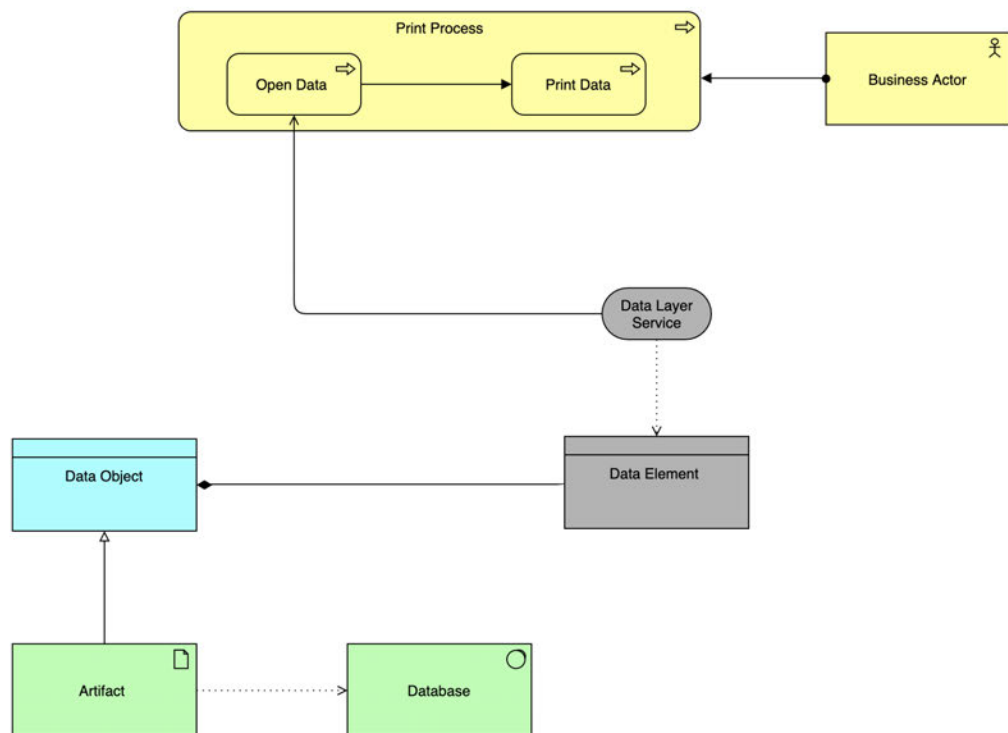


Abbildung 4.11: Medienbruch in Unternehmensprozessen. Quelle: Tom Krause (2020)

5 Umsetzung

5.1 Ausgangssituation

In diesem Kapitel werden die Erweiterungen des Metamodells vorgenommen. Bei den Erweiterungen handelt es sich um Annahmen, die auf den Erkenntnissen aus Kapitel 4 basieren. Nach den Anpassungen des Metamodells soll es folglich möglich sein, allen individuellen Anforderungen aus Kapitel 3.2 gerecht zu werden. Um dies gewährleisten zu können, sind folgende Schritte notwendig:

1.) Das ArchiMate Full Framework (Abbildung 2.7) muss um einen Layer erweitert werden, den Data Layer.

Der Ansatz zur Erweiterung des Full Frameworks ist in Abbildung 5.1 dargestellt. Der neu hinzugefügte Data Layer ist lokal so zuzuordnen, dass alle anderen Layer den Data Layer nutzen können. Somit können zum Beispiel Elemente des Business Layers die Elemente des Data Layers nutzen. Dasselbe gilt für Elemente der restlichen Layer (Strategy, Application, Technology, Implementation). Zusätzlich ist der Data Layer durch seine Anordnung ebenfalls in der Lage, sich weiterhin der Aspekte (Active, Behavior, Passive) in ArchiMate zu bedienen.

2.) Für den Data Layer ist ein eigenes Metamodell zu entwickeln.

Das Metamodell des Data Layers ist in Abbildung 5.2 dargestellt und dient dazu, datenintensive Prozesse in einer Unternehmensarchitektur abbilden zu können. Die Konstruktion des Data Layer Metamodells orientiert sich an dem generischen Metamodell in ArchiMate, welches in Abbildung 2.11 visualisiert wird. Das zusätzlich hinzugefügte Element der Data Location basiert auf der Anforderung 15 aus Kapitel 3.2. Die Data Location dient hierbei den anderen Elementen des Metamodells (Data Internal Behavior Element, Data Event, Data Interface, Data Service, Data Element). Des Weiteren ist das Data Element

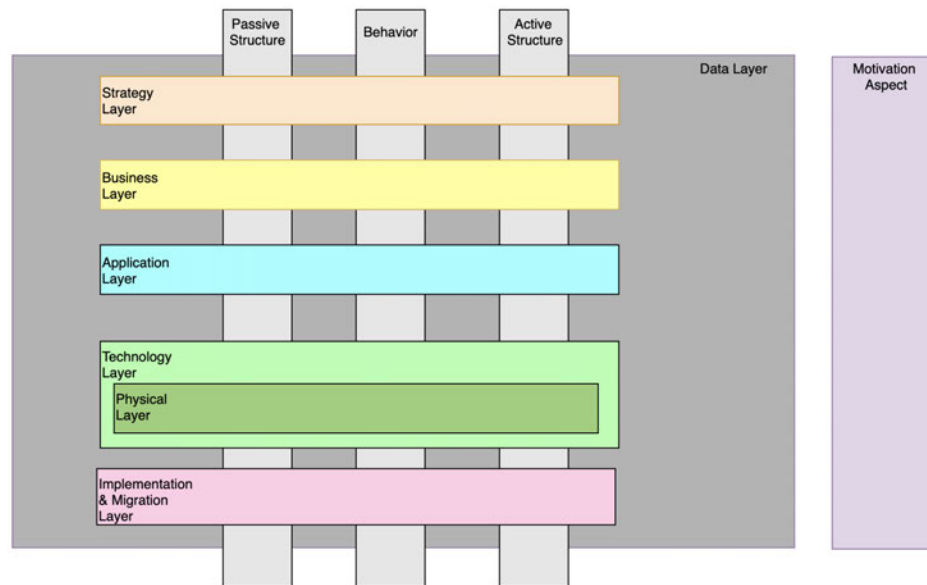


Abbildung 5.1: Erweiterung des ArchiMate Full Frameworks. Quelle: Tom Krause (2020)

der Data Location zugeordnet. Die Abbildung 5.3 liefert eine Übersicht über die Elemente sowie mögliche Instanzen, die durch diesen Layer modelliert werden können. Bei den Instanzen der Elemente handelt es sich um mögliche Beispiele, die auf den Modifikationen der Views und Viewpoints basieren. Folglich stellen sie nur einen Teilauszug dar und können erweitert werden.

3.) Das Data Layer Metamodell muss in das bestehende ArchiMate Metamodell integriert werden.

Das Metamodell des Data Layers ist nun im weiteren Verlauf in das gesamte Metamodell in ArchiMate zu integrieren. Als Ausgangspunkt dienen dazu die bestehenden Beziehungen zwischen den jeweiligen Metamodellen der einzelnen Schichten (Business Layer, Application Layer, Technology Layer). Hierbei handelt es sich um die Cross-Layer Dependencies [Groa].

Abbildung 5.4 stellt den ersten Ausschnitt der Beziehungen zwischen den Schichten dar. Hierbei handelt es sich um die Beziehungen zwischen dem Metamodell des Business Layers (gelb) sowie dem Application Layer (blau). Abbildung 5.5 visualisiert die Beziehungen zwischen dem Metamodell des Business Layers und dem Metamodell des Tech-

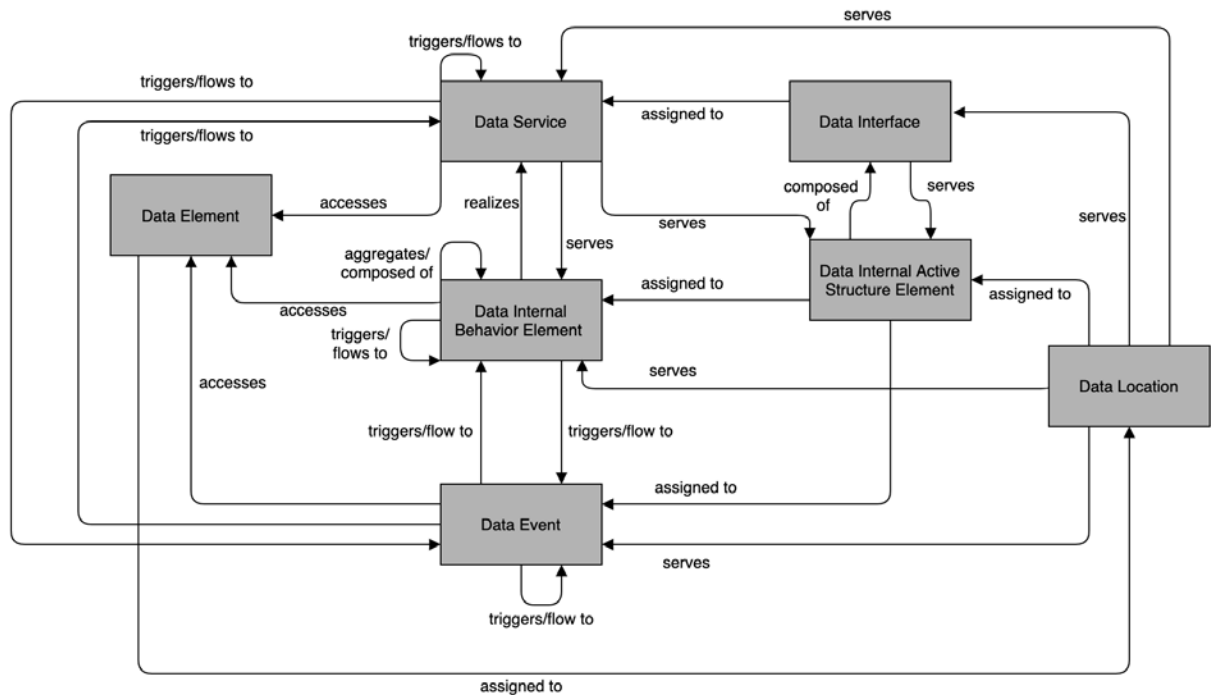


Abbildung 5.2: Data Layer Metamodell. Quelle: Tom Krause (2020)

nology Layers. Des Weiteren werden in Abbildung 5.6 die Beziehungen zwischen dem Application Layer sowie dem Technology Layer (grün) abgebildet.

Bei der Modellierung wird sich hierbei auf die Beziehungen des Kerns von ArchiMate (Business Layer, Application Layer, Technology Layer) konzentriert, da diese hauptsächlich bei der Umsetzung der Anforderungen aus Kapitel 3.2 helfen. Diese gelten als Grundlage und somit ebenfalls nach der Integration des Data Layers. Die Relationen zwischen dem Kern von ArchiMate und dem neu eingeführten Data Layer basieren hierbei auf den Erkenntnissen der vorherigen Kapitel, insbesondere den Anforderungen an das Metamodell.

5.2 Erweiterung des Metamodells

Ähnlich wie bei den Abbildungen 5.4, 5.5, 5.6 werden nun die Beziehungen des Data Layers visualisiert. Hierbei sollen die Beziehungen zwischen dem Data Layer sowie dem Kern von ArchiMate (Business Layer, Application Layer, Technology Layer) abgebildet werden. Die neu hinzugefügten sowie angepassten Relationen (in rot dargestellt) werden im weiteren Verlauf durch Beispiele begründet und evaluiert. Bei den schwarzen Relationen handelt es sich um Beziehungen, welche sich an den Beziehungen der Abbildungen 5.4, 5.5, 5.6 orientieren und hiervon analog abzuleiten sind.

5.2.1 Beziehungen zum Business Layer

Der erste Teilausschnitt der Integration des Data Layers ist in Abbildung 5.7 dargestellt. Hierbei handelt es sich um die Beziehungen zwischen dem Business Layer und dem Data Layer. In Kapitel 3.2 existieren zwei Anforderungen an das Metamodell, die durch die Beziehungen vom Data Layer zum Business Layer visualisiert werden können. Eine der Anforderungen (Anforderung 12) besteht darin, Schwachstellen der Datensicherheit in datenintensiven Prozessen aufzuzeigen. Dies soll durch die Serving-Beziehung zwischen dem Data Service und dem Business Internal Behavior Element gewährleistet werden können. Diese Beziehung wird ausgewählt, da die Medienbrüche oftmals durch Personen in Unternehmensprozessen ausgelöst werden und es sich hierbei um Daten handelt, welche zuvor durch eine Schnittstelle bereitgestellt werden. Folglich können durch diese Relationen Medienbrüche modelliert werden, um Schwachstellen der Datensicherheit abzubilden. Diese Beziehung ist analog von den bestehenden Beziehungen abzuleiten und ist folglich nicht anzupassen.

Die zweite Anforderung (Anforderung 13) besteht darin, die Datenkommunikation zwischen Personen und Systemen in datenintensiven Prozessen abzubilden. Dies soll durch die Serving-Beziehung zwischen dem Data Internal Behavior Element und dem Business Internal Behavior Element gewährleistet sein. Folglich soll es möglich sein, die Datenkommunikation im Data Layer abzubilden, welche dann im Anschluss von den Geschäftsakteuren (Business Actor) in Geschäftsprozessen (Business Layer) benutzt werden kann. Diese Beziehung ist hierfür in der Abbildung 5.7 angepasst.

5.2.2 Beziehungen zum Application Layer

Im zweiten Schritt der Integration sind die Beziehungen zwischen dem Data Layer und dem Application Layer zu ermitteln. Die Abbildung 5.8 visualisiert diese Beziehungen. Die erste hinzugefügte Relation basiert auf der Anforderung vier aus dem Kapitel 3.2. Diese Anforderung besteht darin, eine Echtzeitverarbeitung von Daten durch das Metamodell zu ermöglichen. Bei der Echtzeitverarbeitung sind Applikationen beteiligt, die über eine Schnittstelle erreichbar sind. Bei dieser Schnittstelle handelt es sich um ein Interface, über das die Daten vom Data Layer zu den Applikationen gereicht werden. Folglich wird eine Beziehung benötigt, bei der Funktionen oder Prozesse (Data Internal Behavior Element) des Data Layers vom Application Interface genutzt werden können.

Die zweite Anforderung, die durch die Beziehungen des Data Layers zum Application Layer gewährleistet werden soll, ist Anforderung 14. Diese thematisiert die Datendiversität. Intern soll die Datenvielfalt dadurch abgebildet werden, indem das Data Element (Data Layer) aus dem Data Object besteht (Application Layer). Hierdurch soll die Datenvielfalt durch die einzelnen Datentypen im Data Layer repräsentiert werden.

Eine weitere Anforderung besteht darin, die Lokalität der Daten abzubilden. Dies wird im Metamodell dadurch abgebildet, indem das Data Object (Application Layer) der Data Location (Data Layer) zugeordnet ist.

Die letzte hinzugefügte Beziehung zwischen dem Data Service und dem Application Internal Active Structure Element basiert auf der Abbildung 4.9. Hier dient der Service des Data Layers einer Applikation zur Visualisierung der Daten.

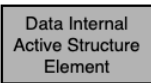

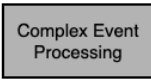
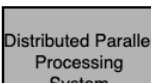
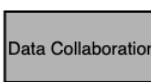
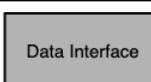
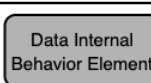
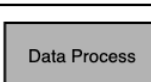
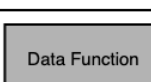
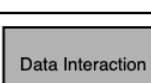
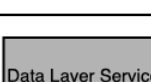
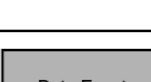
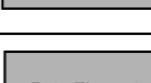
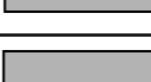

5.2.3 Beziehungen zum Technology Layer

Der letzte Teil der Integration des Data Layers besteht aus den Beziehungen des Technology Layers und dem Data Layer. Diese sind in Abbildung 5.9 visualisiert.

Die erste Beziehung realisiert die Anforderung 15 aus Kapitel 3.2. Hierbei handelt es sich um die Beziehung zwischen dem Technology Object (Technology Layer) und der Data Location (Data Layer). Diese Beziehung gibt an, dass ein Technology Object, wie z.B. ein Artefakt, einer Data Location zugeordnet sein muss, da dieses Artefakt intern in einer Datenbank gespeichert sein kann. Das Gleiche gilt für das Technology Internal

Active Structure Element (Technology Layer). Dieses ist ebenfalls einer Data Location zugeordnet, da es sich hierbei um eine Datenquelle handeln kann.

Die dritte hinzugefügte Beziehung, welche in Abbildung 5.9 dargestellt ist, basiert auf den Anpassungen in Abbildung 4.9. Hierbei dient die Datenquelle aus dem Technology Layer einem System im Data Layer, welches eine verteilte Datenverarbeitung ermöglicht.

Element / Mögliche Instanz	Beschreibung	Notation
Data Internal Active Structure Element	Repräsentiert ein Element, das ein Verhalten erzeugt.	
<i>Data Scientist</i>	Repräsentiert den Data Scientist, der auf der Datenebene für die Verarbeitung und Analyse der Daten zuständig ist.	
<i>Complex Event Processing</i>	Repräsentiert eine Technologie, die intern mit dem Complex Event Processing arbeitet.	
<i>Distributed Parallel Processing System</i>	Repräsentiert eine Technologie, die intern eine verteilte Datenverarbeitung realisiert.	
<i>Data Collaboration</i>	Repräsentiert ein Aggregat aus aktiven Strukturelementen, welches ein Verhalten erzeugt.	
Data Interface	Repräsentiert ein Zugriffselement, an dem Dienste für die Umwelt zugänglich sind.	
Data Internal Behavior Element	Repräsentiert ein Element, das von aktiven Strukturelementen ausgeführt werden kann.	
<i>Data Process</i>	Repräsentiert einen datenintensiven Prozess, in dem Daten vorbereitet und verarbeitet werden.	
<i>Data Function</i>	Repräsentiert eine Funktion, die von einem Datenelement ausgeführt werden kann.	
<i>Data Interaction</i>	Repräsentiert ein Element, das von mehreren Datenelementen ausgeführt wird.	
Data Service	Repräsentiert ein festgelegtes Verhalten der Daten.	
Data Event	Repräsentiert eine Zustandsänderung auf der Datenebene.	
Data Element	Repräsentiert die Daten, an denen das Verhalten ausgeführt wird. Hierbei wird ein Datentyp repräsentiert.	
Data Location	Repräsentiert einen Ort, an dem sich die Elemente befinden oder ausgeführt werden.	
<i>Cloud</i>	Repräsentiert die Cloud, indem sich die Elemente befinden.	

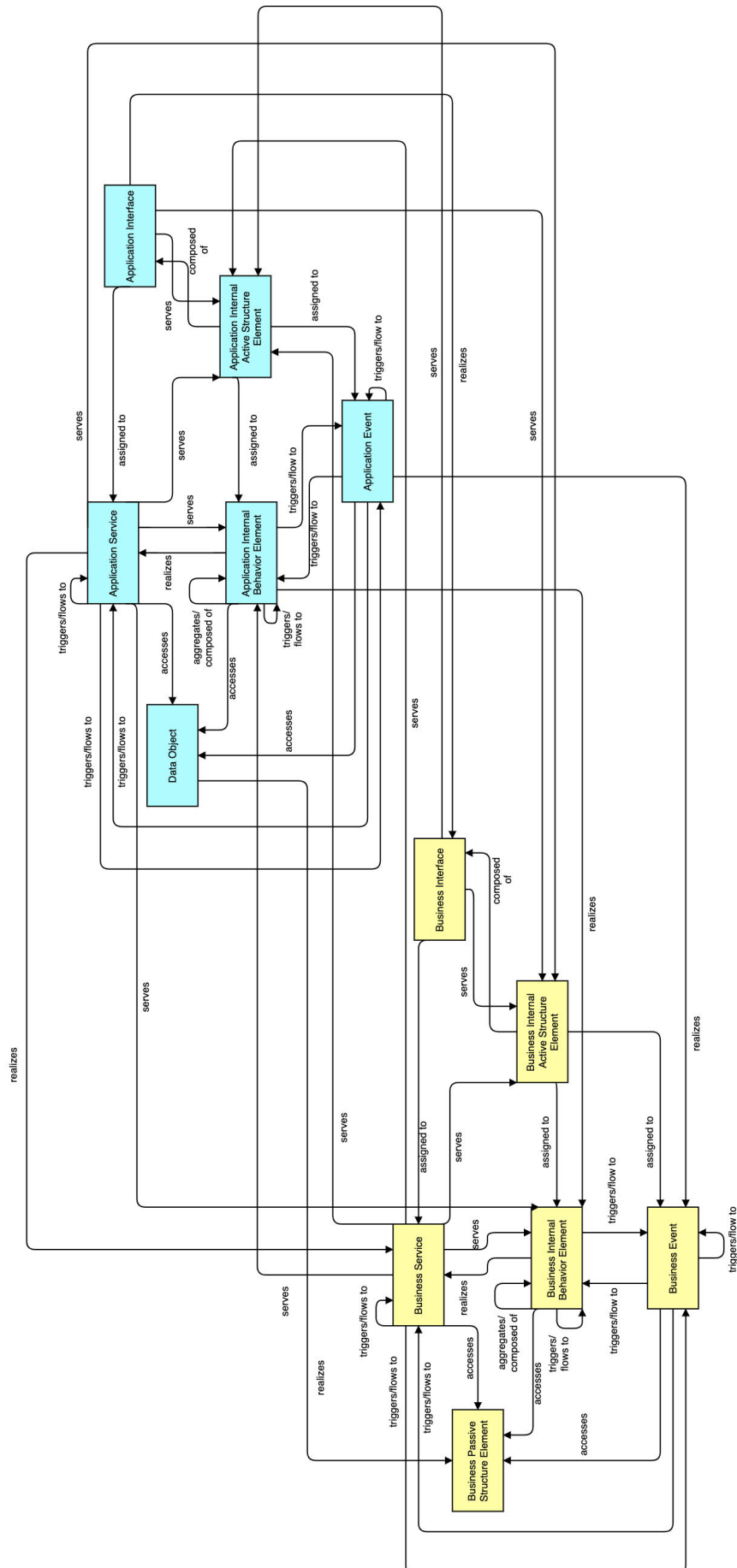


Abbildung 5.4: Beziehungen der Metamodelle zwischen dem Business Layer und dem Application Layer. Quelle: In Anlehnung an [Groa]

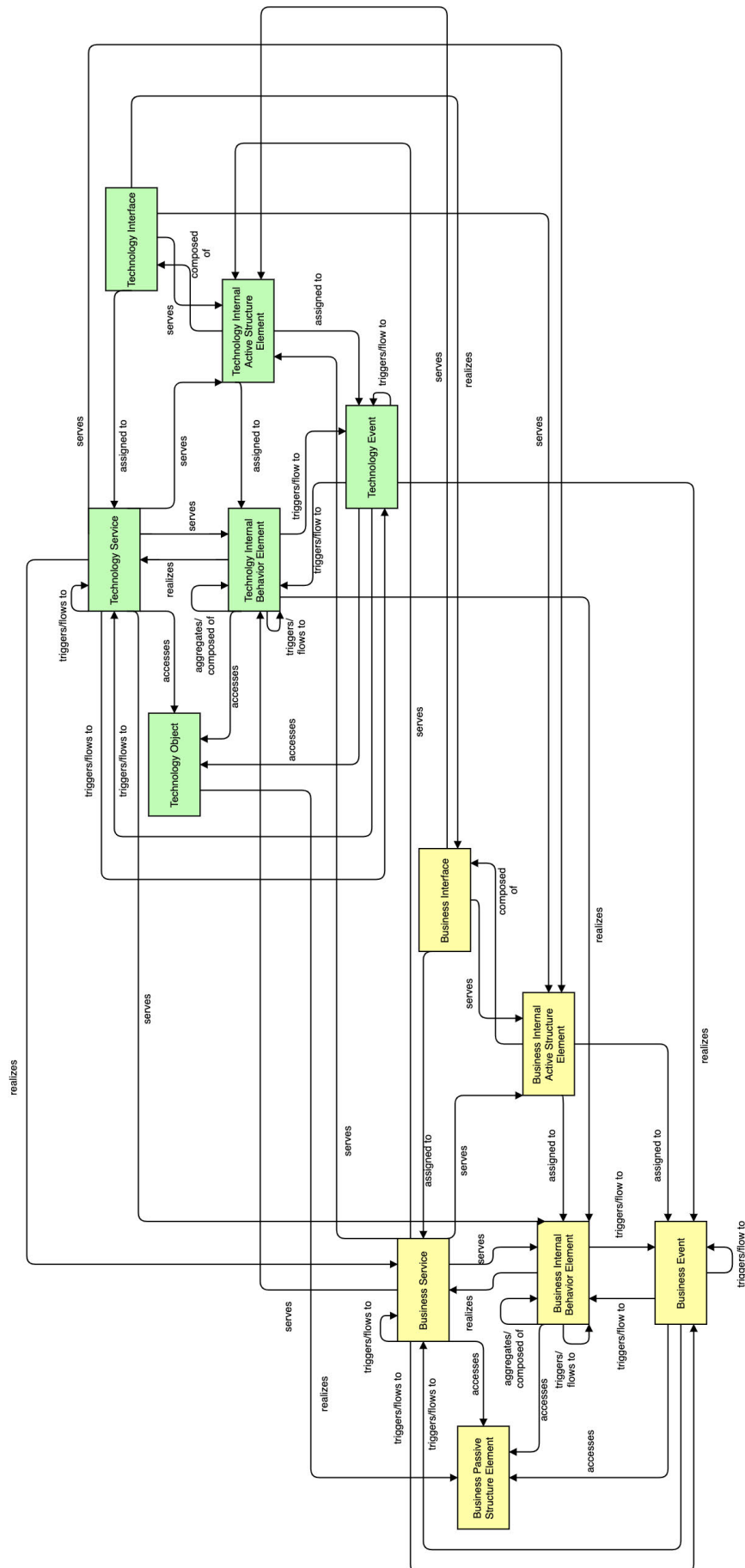


Abbildung 5.5: Beziehungen der Metamodelle zwischen dem Business Layer und dem Technology Layer. Quelle: In Anlehnung an [Groa]

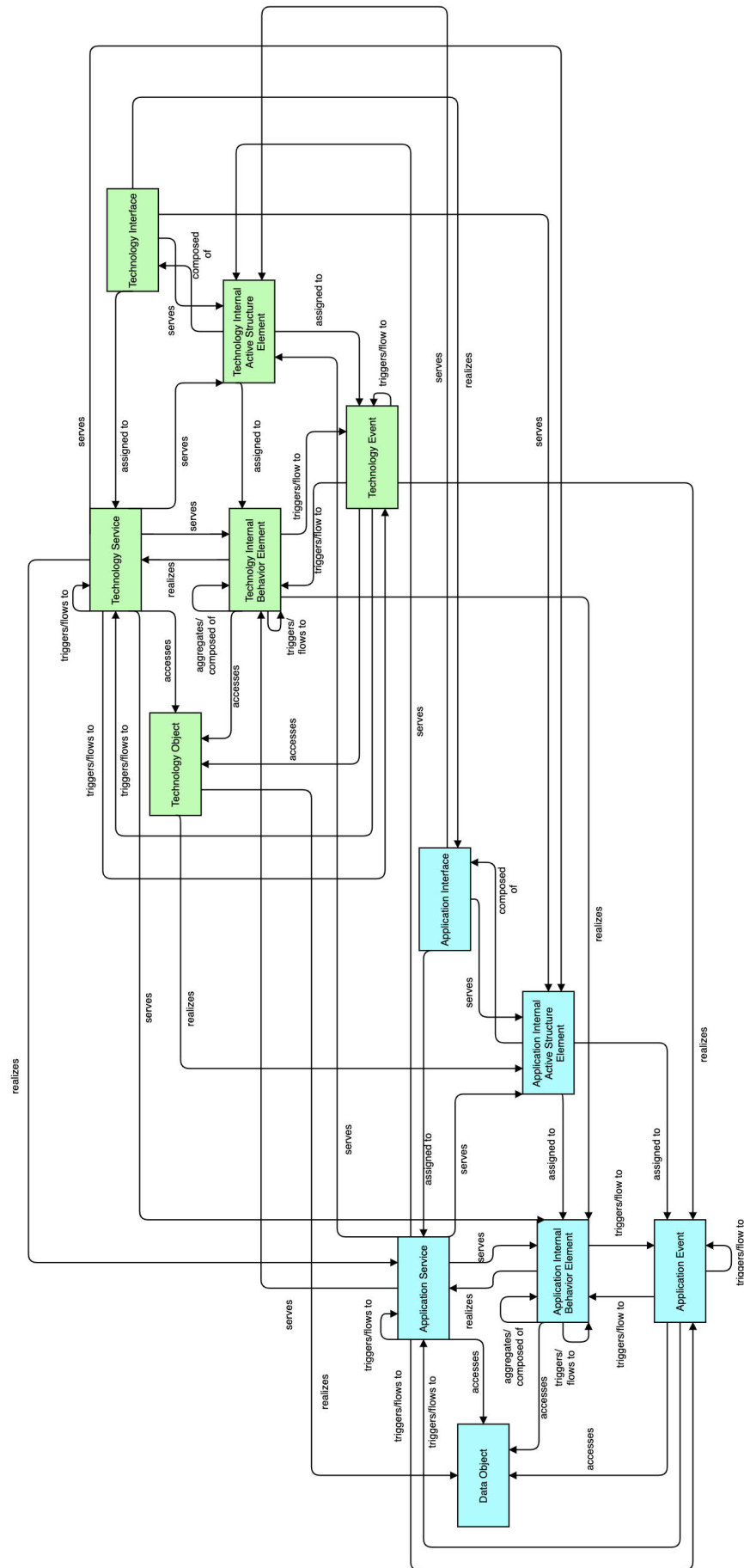


Abbildung 5.6: Beziehungen der Metamodelle zwischen dem Application Layer und dem Technology Layer. Quelle: In Anlehnung an [Groa]

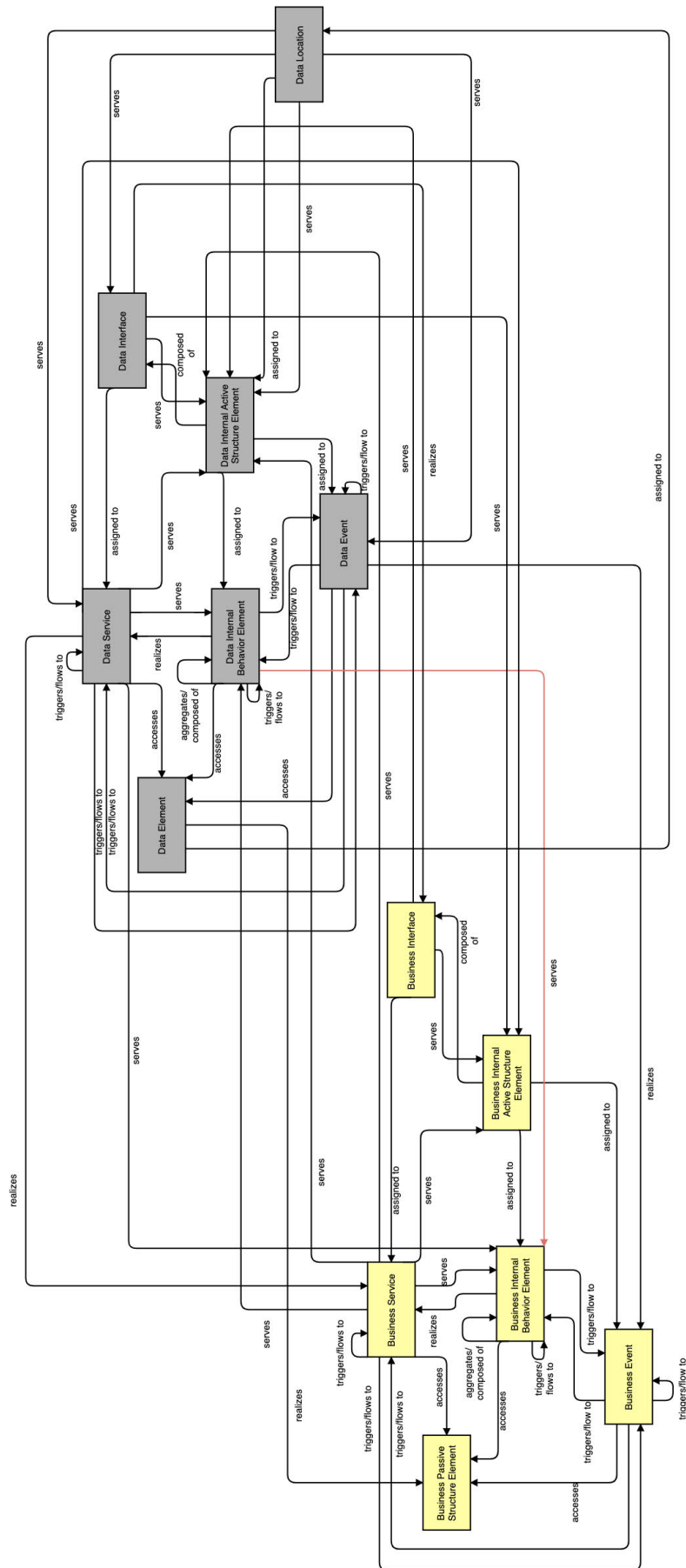


Abbildung 5.7: Beziehungen zwischen dem Data Layer und dem Business Layer. Quelle: Tom Krause (2020)

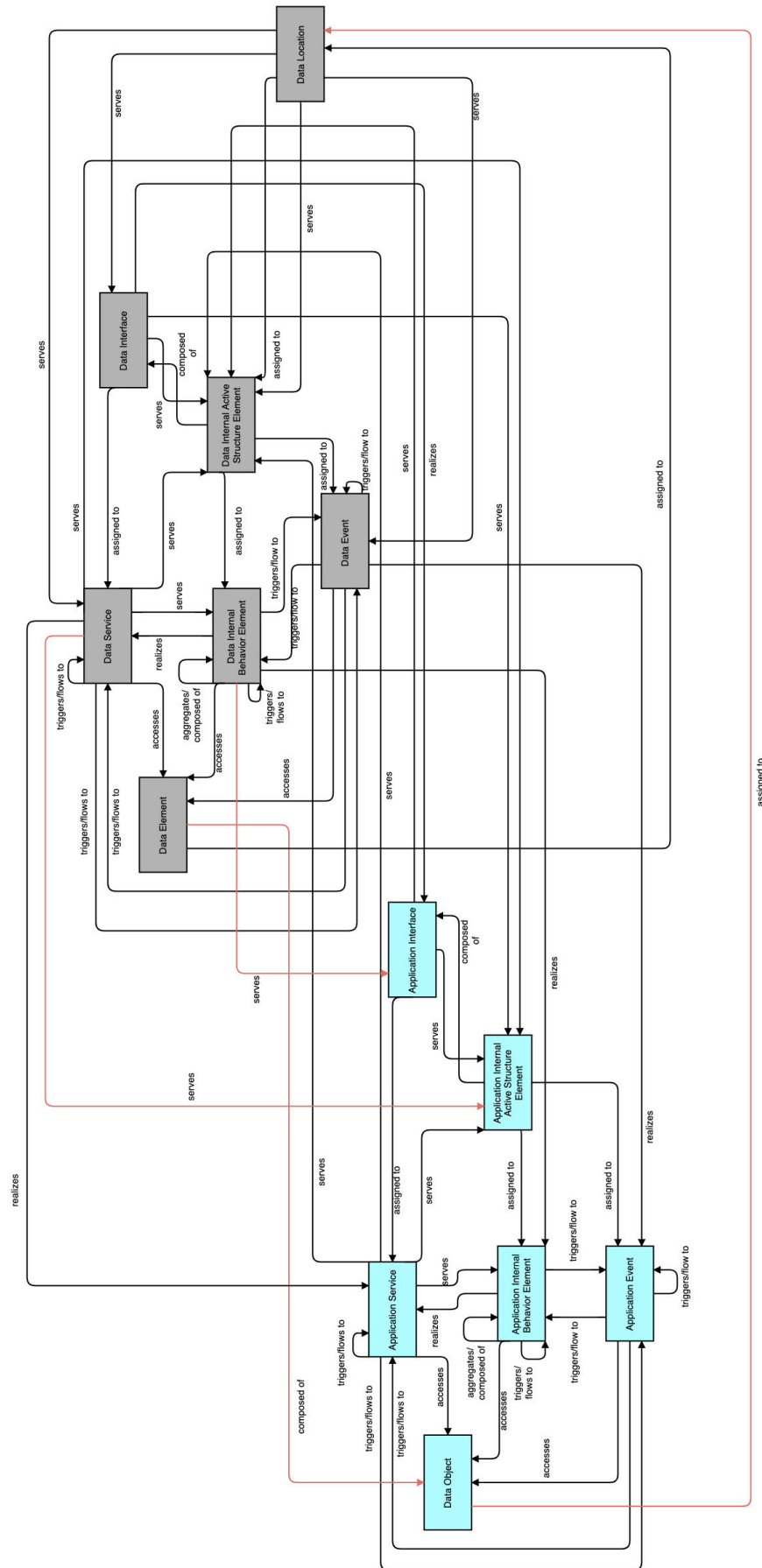


Abbildung 5.8: Beziehungen zwischen dem Data Layer und dem Application Layer. Quelle: Tom Krause (2020)

6 Evaluation

In diesem Kapitel wird das erweiterte Metamodell evaluiert. Für diese Evaluation werden Modellbeispiele entworfen. Anschließend ist zu untersuchen, ob die Beispiele durch das Metamodell realisierbar sind. Hierzu wird ein Teilauszug der Bestandteile (Relationen und Elemente) vom erweiterten Metamodell untersucht. Dieser Teilauszug besteht aus den neu hinzugefügten Bestandteilen (Relationen, Elemente) sowie denen, die analog abgeleitet wurden. Das erweiterte Metamodell wird dahingehend untersucht, ob die Elemente und Relationen in konkreten Modellen dazu dienen, die verbleibenden Anforderungen aus Kapitel 4 zu realisieren. Bei den Modellbeispielen handelt es sich um Szenarien, die auf einem Kassensystem basieren. Abbildung 6.1 stellt eine Übersicht dieses Systems dar. Das System ist ein Kassensystem, welches Cloud-basiert arbeitet und eine Zentralverwaltung von Kassen ermöglicht. Zusätzlich ist das Kassensystem dazu in der Lage, Auswertungen in Echtzeit zu ermöglichen. Des Weiteren ist das System dazu fähig, eine verschlüsselte Datenübertragung zu gewährleisten [Gmb20].

6.1 Modellierung von Szenarien

Im Folgenden werden Szenarien des Kassensystems modelliert, bei denen datenintensive Prozesse ein Bestandteil sind. Die modellierten Szenarien versuchen, die verbleibenden individuellen Anforderungen aus dem Kapitel 3.2 umzusetzen.

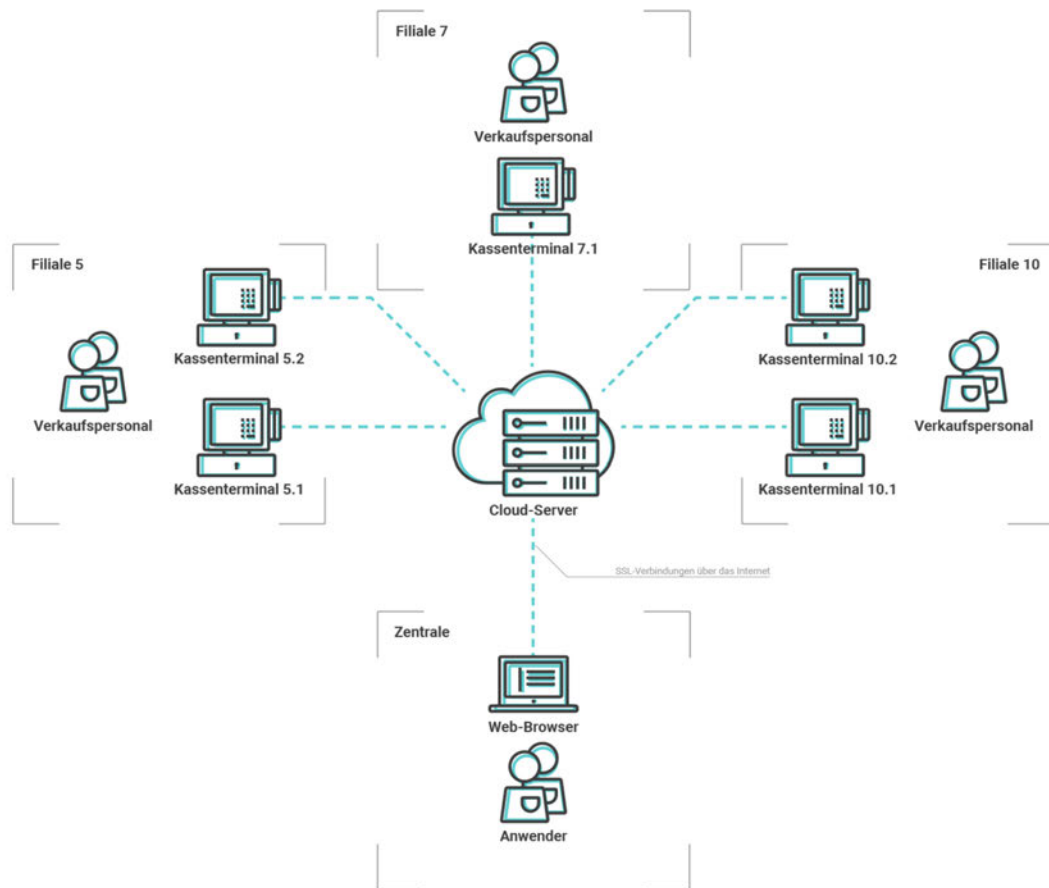


Abbildung 6.1: Systemübersicht. Quelle: [Gmb20]

6.1.1 Anmeldung des Systems

Das erste Szenario demonstriert die Anmeldung einer Kasse, welche hierbei als Applikation modelliert wird. Dieser Vorgang ist in Abbildung 6.2 dargestellt. Hierbei erhält die Kasse Stammdaten von der Zentrale, damit sie anschließend in einer Filiale vom Verkaufspersonal benutzt werden kann. Als Erstes werden die Daten aus der PostgreSQL-Datenbank extrahiert, welche dem Element der Datenherkunft (Data Location) zugeordnet ist. Diese Rohdaten aus der Kassentabelle realisieren die Kassendaten. Die Kassendaten bestehen intern aus dem Data Element (Stammdaten). Bei den Stammdaten handelt es sich um die abgebildeten Spezialisierungen, welche ebenfalls der Datenherkunft zuzuordnen sind. Die Stammdaten sowie die Spezialisierungen dieser Stammdaten sollen die Datendiversität durch verschiedene Datentypen repräsentieren. Im Anschluss werden

die Stammdaten der Kasse über den Data Layer Service (Stammdaten bereitstellen) zur Kassenapplikation gereicht. Folglich können die Daten im Prozess der Kassenanmeldung vom Verkaufspersonal genutzt werden. Dieses Szenario zeigt, dass das erweiterte Metamodell die Anforderung 14 sowie die Anforderung 15 aus Kapitel 3.2 abbilden kann. Bei den rot dargestellten Beziehungen handelt es sich um neu hinzugefügte Relationen aus der Integration des Data Layers. Diese Relationen sind somit in einer konkreten Modellierung realisierbar und dienen als Unterstützung zur Umsetzung der zuvor beschriebenen Anforderungen.

6.1.2 Datenbereitstellung

In diesem Szenario (Abbildung 6.3) wird die Bereitstellung der Kassendaten für das Finanzamt modelliert. Die Kassendaten, welche ursprünglich aus einer Datenbank extrahiert wurden, sind dabei auf einen Stick zu exportieren. Dieser Stick dient dem Finanzamt dazu, die Kassendaten zu prüfen. Hierbei handelt es sich z.B. um Tagesabschlüsse, Bons oder Identifier. Die Kassendaten werden über den Data Layer Service bereitgestellt. Im Anschluss können die Daten geöffnet und auf einen Stick übertragen werden. Bei der rot gezeichneten Relation zwischen dem Data Service und dem Prozess der Datenübertragung (Business Internal Behavior Element) handelt es sich um eine Relation aus der Abbildung 5.7. Diese Relation dient in diesem Szenario dazu, dass die Daten vom Data Layer durch die Mitarbeiter des Finanzamtes exportiert werden können. Des Weiteren kennzeichnet diese Relation eine potenzielle Schwachstelle in der Datensicherheit (Anforderung 12). Dies ist der Fall, da es sich hierbei um einen Medienbruch handelt, weil die Übertragung von Personen durchgeführt wird. Die restlichen Relationen, welche in rot dargestellt werden, sind analog vom ersten Szenario zu übernehmen. Die Abbildung 6.3 zeigt, dass die Relation zwischen dem Data Service und dem Business Internal Behavior Element modelliert werden kann und dazu dient, Schwachstellen der Datensicherheit zu kennzeichnen.

6.1.3 Datenflüsse zu Nachbarsystemen

Die Abbildung 6.4 visualisiert ein Beispiel, welches einen Datenfluss zwischen Systemen modelliert. Das Szenario hat den Zweck, zu überprüfen, ob das erweiterte Metamodell eine Datenkommunikation (Anforderung 13) abbilden kann. Hierbei werden Daten aus der

Cloud zu einer externen Cloud exportiert. Bei den Daten handelt es sich um Arbeitszeiten und Identifier. Die ActiveMQ ist eine Message, welche die Schnittstelle zur externen Cloud darstellt und den Datenaustausch technisch realisiert. Bei den roten Beziehungen handelt es sich erneut um die Relationen, welche bereits in den vorherigen Szenarien modelliert wurden und hierfür analog zu übernehmen sind. Die blaue Beziehung wird in dem Szenario benötigt. Diese kann jedoch durch das erweiterte Metamodell nicht abgebildet werden. Folglich zeigt dieses Szenario, dass das Metamodell nicht komplett dafür geeignet ist, die Datenkommunikation in datenintensiven Prozessen zu visualisieren. Das erweiterte Metamodell kann der Anforderung 13 aus Kapitel 3.2 somit nicht gerecht werden.

6.1.4 Pflege eines Systemverbunds

Das vierte Szenario in Abbildung 6.5 zeigt die Pflege des Kassenverbunds. Die Kassen werden hier als Applikationen repräsentiert. Bei der Pflege des Kassenverbunds werden die einzelnen Kassen in Echtzeit mit neuen Stammdaten aktualisiert. Die Stammdaten stammen aus der Unternehmenszentrale. Hierbei handelt es sich um Berichte (Umsatzberichte, Warenfrequenzen, Abverkäufe) und um Retouren (Retourenquoten, Warenrücknahmen, Retourenmengenkontrollen). Technisch realisiert wird die Echtzeitverarbeitung durch eine REST API, welche einen Eventstream anbietet. Die rote Relation zwischen dem Application Interface (REST API Eventstream) und dem Data Internal Behavior Element (Datenkommunikation) stammt aus dem erweiterten Metamodell (Kapitel 5). Die restlichen in rot gezeichneten Beziehungen gelten analog zu den Beispielen davor. Dieses Beispiel zeigt, dass das erweiterte Metamodell dazu dient, eine Datenverarbeitung in Echtzeit (Anforderung 4) in einem konkreten Modell zu modellieren. Dies ist der Fall, da in Abbildung 5.8 eine neue Relation zwischen dem Data Internal Behavior Element und dem Application Interface hinzugefügt wurde.

6.2 Bewertung

Die Modellierung der Beispielszenarien lässt folgende Schlussfolgerungen zu:

Das erweiterte Metamodell, welches durch einen Data Layer sowie die zugehörigen Relationen angepasst wurde, ist größtenteils dazu fähig, datenintensive Prozesse in konkreten

Modellen abzubilden. Dies ist der Fall, da fast alle verbliebenen individuellen Anforderungen, welche in Kapitel 4 ermittelt wurden, durch die neuen Relationen im Metamodell realisiert werden konnten. Allerdings zeigt sich, dass Optimierungsbedarf bei dem erweiterten Metamodell besteht. In der Abbildung 6.4 fehlt eine Relation, die das Metamodell beinhalten muss, damit das Szenario vollständig modelliert werden kann. Folglich müsste das Metamodell weiter angepasst werden und diese notwendige Beziehung abbilden.

Außerdem fällt bei den Szenarien auf, dass sich der Aufbau der konkreten Modelle häufig ähnelt. Dieser ähnliche Aufbau ergibt sich dadurch, dass die benutzten Elemente des Data Layers oftmals nur in Relation mit zwei Ebenen gesetzt wurden, dem Application Layer sowie dem Technology Layer. Folglich sollte das Metamodell so angepasst werden, dass die Elemente aus dem Data Layer besser in Beziehung zum Business Layer stehen. Dies könnte im Rahmen einer weiteren Arbeit genauer untersucht werden.

Hinzu kommt, dass die Beispiele in den Abbildungen 6.2, 6.3, 6.4, 6.5 oftmals Relationen visualisieren, welche als redundant angesehen werden können. Ein Beispiel für diese Redundanzen wären die Artefakte, welche der Datenherkunft zugeordnet sind sowie die Datenbanken, aus denen diese Artefakte stammen. Diese Datenquellen werden ebenfalls aufgrund des erweiterten Metamodells der Datenherkunft (Data Location) zugewiesen. Aufgrund dieser Redundanzen lässt sich die Schlussfolgerung ziehen, dass einige Relationen im erweiterten Metamodell nicht benötigt werden.

Des Weiteren wurden die visualisierten Geräte (Kassen) nur als Kassenapplikationen dargestellt. Man könnte diese Kassen ebenso als Geräte im Technology Layer modellieren. Folglich könnte man im Rahmen einer weiteren Arbeit die benötigten Relationen für ein erweitertes Metamodell ermitteln.

Zusammenfassend ist während der Modellierung der Beispiele deutlich geworden, dass das angepasste und erweiterte Metamodell dazu in der Lage ist, datenintensive Prozesse abzubilden. Jedoch gibt es noch Schwachstellen des Metamodells, die man weiter anpassen könnte. Zusätzlich basiert die Evaluation des Metamodells auf ausgesuchten Anforderungen. Somit könnte man die Anforderungen an ein Metamodell erweitern, welche man anhand von einheitlichen und objektiven Bewertungskriterien anschließend evaluiert.

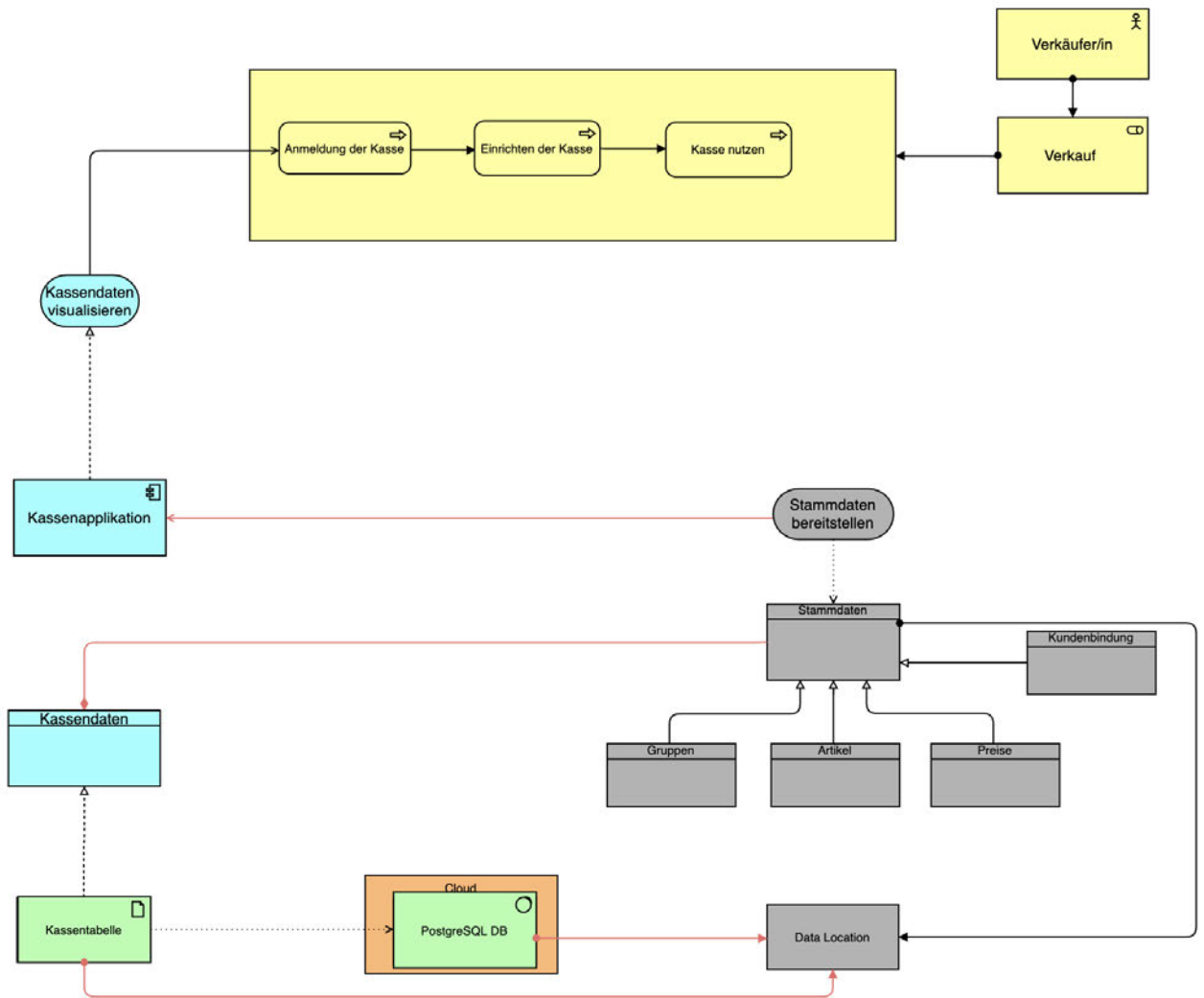


Abbildung 6.2: Anmeldung einer Kasse. Quelle: Tom Krause (2020)

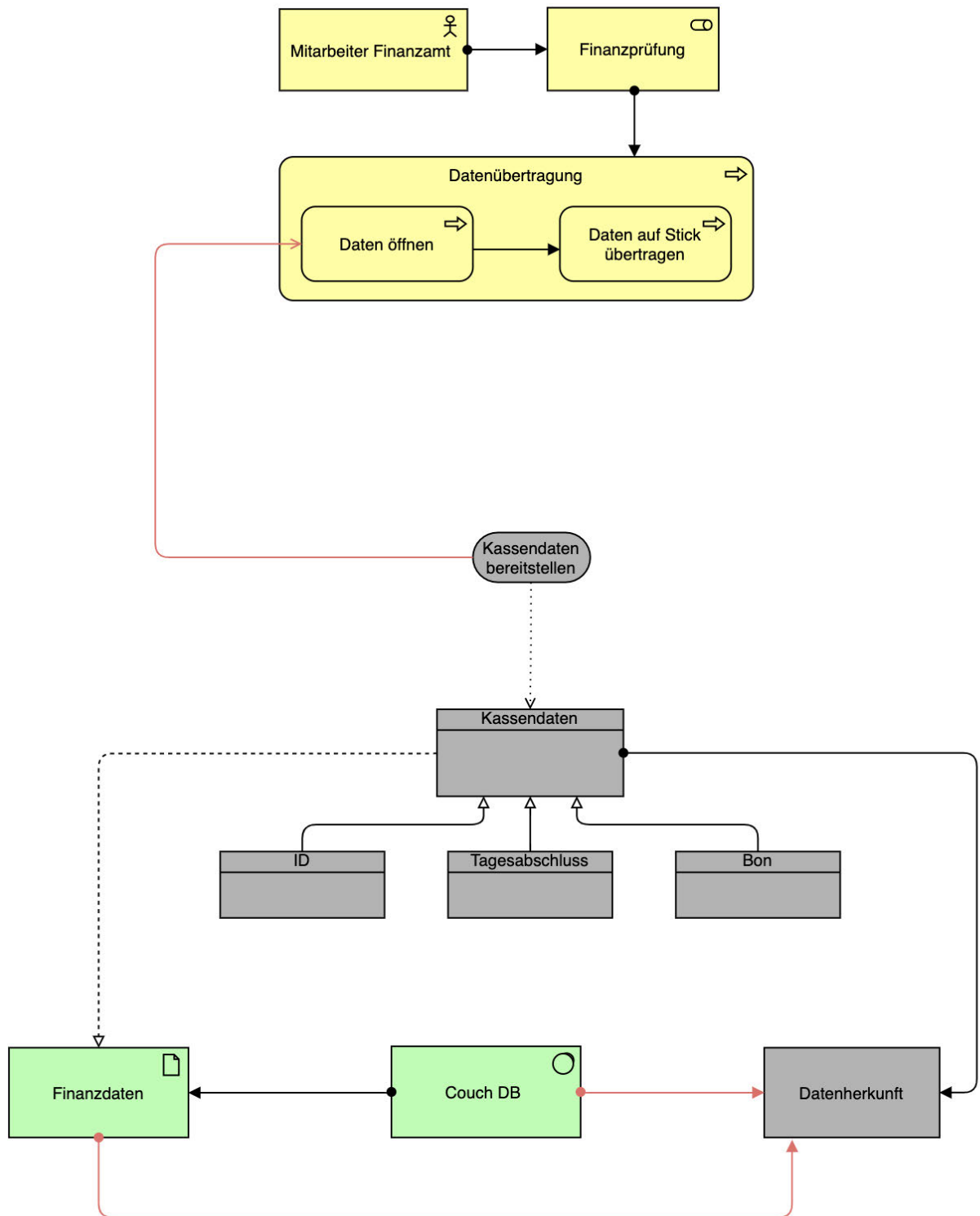


Abbildung 6.3: Datenbereitstellung für das Finanzamt. Quelle: Tom Krause (2020)

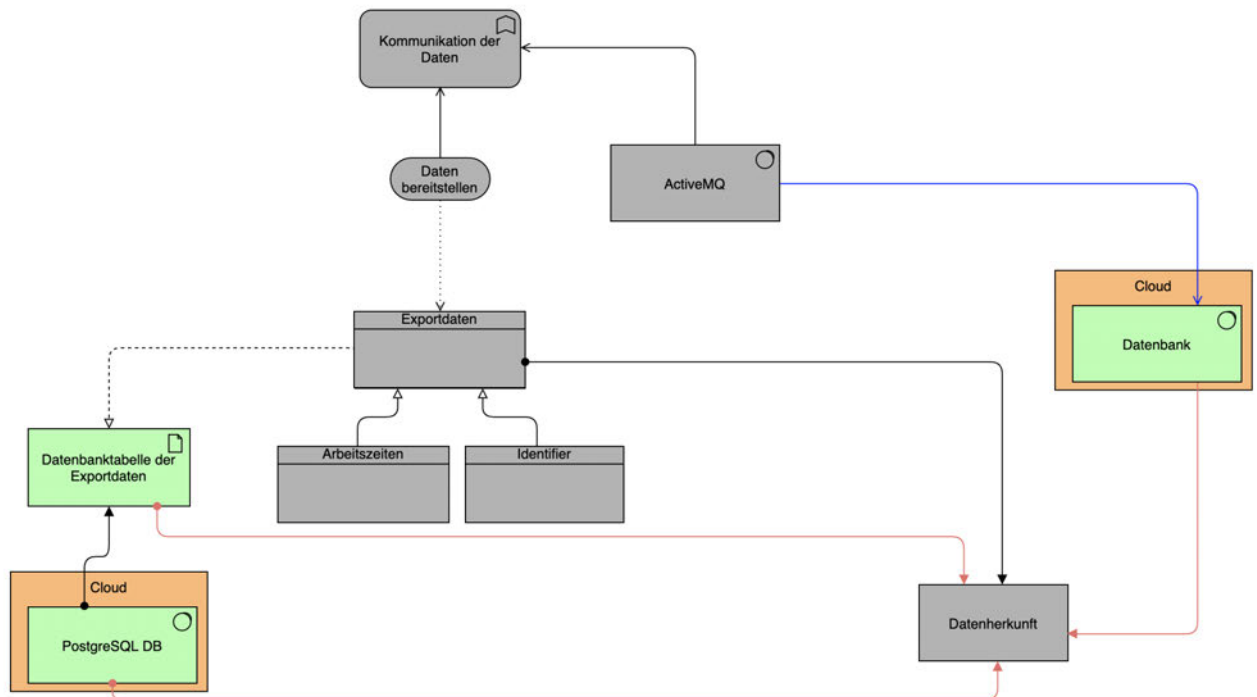


Abbildung 6.4: Datenflüsse zu Nachbarsystemen. Quelle: Tom Krause (2020)

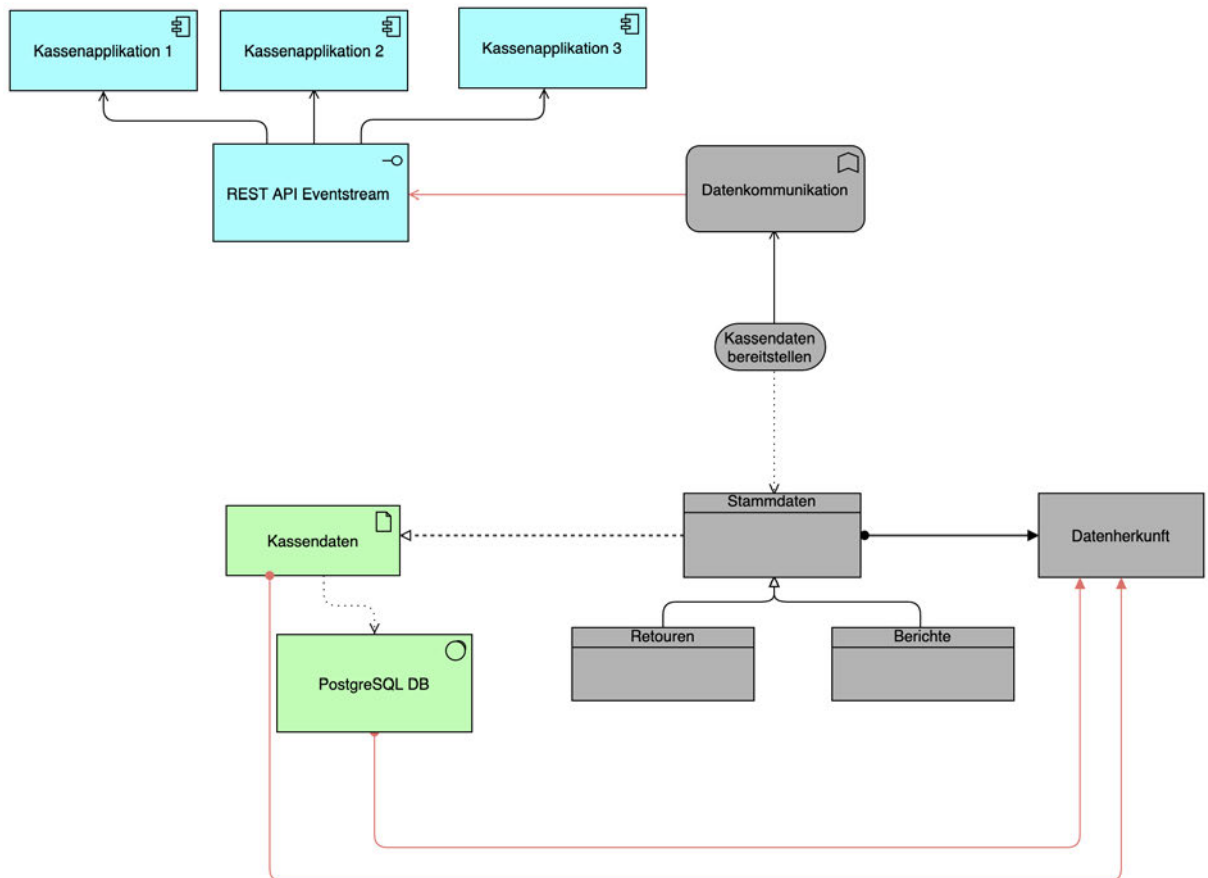


Abbildung 6.5: Pflege des Kassenverbands. Quelle: Tom Krause (2020)

7 Zusammenfassung

Im Rahmen dieser Arbeit wurde die Frage untersucht, ob das bestehende Metamodell in ArchiMate geeignet ist, um datenintensive Prozesse in Unternehmen abzubilden.

Um diese Frage beantworten zu können, wurde zunächst der Stand der Forschung beschrieben. Hierbei bestand die Intention darin, zwei relevante Themengebiete für die zuvor beschriebene Fragestellung genauer zu betrachten. Bei den Themengebieten handelte es sich um Big Data, Enterprise Architecture Management inklusive des Prinzips der Metamodellierung.

Damit die Voraussetzungen zur Abbildung datenintensiver Prozesse ermittelt werden konnten, wurden zunächst der Begriff Big Data definiert sowie dessen Charakteristiken. Im Anschluss wurden dann im Kapitel 2.1 die Vorteile von Big Data sowie die Erfolgsfaktoren von Projekten, in denen Big Data genutzt wird, beschrieben. Des Weiteren wurden die Technologien von Big Data beschrieben. Im nächsten Schritt sind dann die Probleme und Herausforderungen von Big Data ermittelt worden.

In Kapitel 2.2 wurde dann das Enterprise Architecture Management definiert sowie dessen Vorteile beschrieben. Zusätzlich war es wichtig, die Unterstützung dieses Themengebiets für Big Data darzustellen.

Um die Forschungsfrage dieser Arbeit zu klären, wurden die Modellierungssprache ArchiMate sowie deren Bestandteile erklärt. Auf Basis dieser Inhalte ist das Prinzip der Metamodellierung dargestellt worden.

Die gewonnenen Erkenntnisse aus Kapitel 2 dienen im weiteren Verlauf dazu, Anforderungen an ein Metamodell (Kapitel 3) zur Abbildung datenintensiver Prozesse zu definieren. Diese Anforderungen waren in allgemeine sowie individuelle Anforderungen zu unterteilen.

Die Anforderungen aus Kapitel 3 bildeten die Grundlage für das Kapitel 4. In diesem Kapitel wurden Viewpoints und Views entwickelt. Hierbei bestand die Intention, zu untersuchen, ob das bestehende Metamodell in ArchiMate dazu geeignet ist, den Anforderungen aus Kapitel 3 gerecht zu werden. Der Fokus wurde dabei auf die zuvor ermittelten individuellen Anforderungen gelegt. Die Konzeptionen aus Kapitel 4 haben dabei gezeigt, dass das Metamodell in ArchiMate nicht dazu geeignet ist, datenintensive Prozesse abzubilden. Dies war der Fall, da nicht alle individuellen Anforderungen aus Kapitel 3.2 in die Konzeptionen aus dem Kapitel 4 integriert werden konnten. Folglich brachte das Kapitel 4 die Erkenntnis, das Metamodell in ArchiMate anzupassen und zu erweitern.

In Kapitel 5 sind die Anpassungen und Erweiterungen des Metamodells realisiert worden. Hierbei wurde das Metamodell in ArchiMate um einen Layer erweitert, den Data Layer. Dabei bestand die Herausforderung darin, den Data Layer in das bestehende Metamodell zu integrieren. Um diese Integration umzusetzen, wurden Relationen im Metamodell angepasst und erweitert.

Im Anschluss wurde das erweiterte Metamodell evaluiert. Diese Evaluation basierte auf einem Kassensystem, welches in Kapitel 6 genauer beschrieben wurde. Die Erkenntnis dieses Kapitels bestand darin, dass das erweiterte Metamodell größtenteils dazu geeignet ist, datenintensive Prozesse abzubilden. Die Schwachstellen sowie die Einschränkungen dieses Metamodells wurden dabei ebenfalls ermittelt.

7.1 Fazit

Die in Kapitel 4 erarbeiteten Konzeptionen zeigen, dass das bestehende Metamodell in ArchiMate nicht dazu geeignet ist, datenintensive Prozesse in Unternehmen abzubilden. Diese Arbeit liefert Ansätze zur Erweiterung des Metamodells. Der Entwicklungsprozess dieser Arbeit lässt folgende Schlussfolgerungen zu:

Das Kapitel 4 und dessen dargestellten Konzeptionen belegen, dass ArchiMate als Modellierungssprache bereits ein Metamodell besitzt, welches in seiner Syntax und Semantik eine umfangreiche Grundlage darstellt, um Unternehmensarchitekturen zu modellieren. Dies ist der Fall, da der Großteil der formulierten Anforderungen (Kapitel 3) durch das bestehende Metamodell modelliert werden konnte. Außerdem lässt sich diese Erkenntnis dadurch belegen, dass das existierende Metamodell als Grundlage zur Erweiterung genutzt wurde. Folglich wurden die Bestandteile (Elemente, Relationen) teilweise analog

übernommen. Eine weitere Erkenntnis besteht jedoch darin, dass das bestehende Metamodell in ArchiMate angepasst werden muss, um datenintensive Prozesse abbilden zu können. Die im Rahmen dieser Arbeit konzipierten Erweiterungen des Metamodells sind größtenteils in der Lage, datenintensive Prozesse abzubilden. Das Kapitel 6 belegt das anhand der vorgenommenen Untersuchungen.

Weiterhin zeigt sich jedoch, dass das erweiterte Metamodell Schwachstellen besitzt und Optimierungspotenziale aufweist. Diese Erkenntnis basiert darauf, dass das konzipierte Metamodell nur auf Basis von kleinen Beispielszenarien evaluiert wurde. Die Eignung des Metamodells ist somit nur auf einen kleinen Rahmen beschränkt und nicht in Unternehmen erprobt.

Des Weiteren zeigt diese Arbeit, dass Big Data Probleme und Herausforderungen für Unternehmen impliziert. Dies ist ein Indikator dafür, dass der Bedarf an Unternehmensarchitekturen, welche für datenintensive Prozessabläufe ausgelegt sind, steigt. Um eine Orientierung für diese Unternehmensarchitekturen zu schaffen, werden Modelle sowie deren Metamodelle benötigt. Die Metamodelle müssen in der Lage sein, die notwendigen Bestandteile zur Modellierung von datenintensiven Szenarien zu beinhalten. Diese Arbeit liefert einen konzeptionellen Ansatz für solch ein Metamodell und dessen Bestandteile (Elemente, Relationen).

7.2 Ausblick

Das entwickelte Metamodell bildet eine Grundlage zur Abbildung datenintensiver Prozesse, welche als Basis für weitere Arbeiten dienen kann. Das erweiterte Metamodell konzentriert sich in der Konstruktion auf den Kern von ArchiMate (Business Layer, Application Layer, Technology Layer). Im Rahmen weiterer Arbeiten könnte man bei der Erweiterung des Metamodells zudem das gesamte ArchiMate Full Framework berücksichtigen inklusive weiterer Ebenen (Implementation und Migration Layer, Strategy Layer) sowie den Motivation Aspect.

Zusätzlich könnte man das entwickelte Metamodell in der Praxis erproben. Hierzu wäre es möglich, dass man verschiedene Unternehmen sowie deren datenintensive Prozesse betrachtet. Folglich hätte man einen Eindruck, inwiefern das entwickelte Metamodell in komplett unterschiedlichen datenintensiven Szenarien praxisnah für die Modellierung realisierbar ist.

Ein weiterer Ansatz zur Forschung wäre die Evaluation dieses Metamodells mit echten Daten. Diese Daten sollen sich durch ihre diverse Beschaffenheit und großen Mengen auszeichnen. Hierbei könnte man ein Konzept entwickeln, welches das Metamodell z.B. mit einem Ansatz der Künstlichen Intelligenz evaluiert.

Des Weiteren könnte man die Konzepte dieser Arbeit mittels einer Programmiersprache implementieren. Hierbei wäre es möglich, die hinzugefügten Elemente in dem erweiterten Metamodell als Entitäten zu betrachten. Diese würden dann als Orientierung für eine Programmierung dienen. Folglich könnte man das Metamodell durch eine Implementation in einer objektorientierten Programmiersprache evaluieren.

Es lässt sich daher festhalten, dass zusätzliche Forschungen, bestehend aus Weiterentwicklungen und Evaluationen des Metamodells, weitere Fortschritte für die Abbildung datenintensiver Prozesse implizieren würden.

Literaturverzeichnis

- [AAKE13] Turki Alsudiri, Wafi Al-Karaghoul, and Tillal Eldabi. Alignment of large project management process to business strategy: A review and conceptual framework. *Journal of Enterprise Information Management*, 26(5):596–615, 2013.
- [ABB⁺11] D Agrawal, P Bernstein, E Bertino, S Davidson, U Dayal, M Franklin, J Gehrke, L Haas, A Halevy, J Han, et al. Challenges and opportunities with big data. purdue university. Technical report, Cyber Center Technical Reports, 2011.
- [AK03] Colin Atkinson and Thomas Kuhne. Model-driven development: a metamodeling foundation. *IEEE software*, 20(5):36–41, 2003.
- [ASC17] A. Abbasi, S. Sarker, and R. Chiang. Big data research in information systems: Toward an inclusive research agenda. *Journal of the Association for Information Systems*, 2017.
- [ASML12] Frederik Ahlemann, Eric Stettiner, Marcus Messerschmidt, and Christine Legner. *Strategic enterprise architecture management: challenges, best practices, and future developments*. Springer Science & Business Media, 2012.
- [BB12] Jean Burgess and Axel Bruns. Twitter archives and the challenges of "big social data" for media and communication research. *M/C Journal*, 15(5), 2012.
- [BC12] Danah Boyd and Kate Crawford. Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. *Information, communication & society*, 15(5):662–679, 2012.
- [Bre00] Eric A Brewer. Towards robust distributed systems. In *PODC*, volume 7, pages 343477–343502. Portland, OR, 2000.

- [Cam14] Brian H Cameron. The need for enterprise architecture for enterprise-wide big data. *ISJLP*, 10:827, 2014.
- [CCS12] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. Business intelligence and analytics: From big data to big impact. *MIS quarterly*, pages 1165–1188, 2012.
- [CK15] I. Constantiou and J. Kallinikos. New games, new rules: big data and the changing context of strategy. *Journal of Information Technology*, 2015.
- [Cle09] Eric K Clemons. Business models for monetizing internet applications and web sites: Experience, theory, and predictions. *Journal of Management Information Systems*, 26(2):15–41, 2009.
- [CZ14] CL Philip Chen and Chun-Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information sciences*, 275:314–347, 2014.
- [D⁺06] Thomas H. Davenport et al. Competing on analytics. *Harvard business review*, 84:98, 2006.
- [Dav14] Thomas Davenport. *Big data at work: dispelling the myths, uncovering the opportunities*. Harvard Business Review Press, 2014.
- [DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- [DMGG15] Andrea De Mauro, Marco Greco, and Michele Grimaldi. What is big data? a consensual definition and a review of key research topics. In *AIP conference proceedings*, volume 1644, pages 97–104. American Institute of Physics, 2015.
- [DR12] Paul C Dinsmore and Luiz Rocha. *Enterprise project governance: a guide to the successful management of projects across the organization*. Amacom, 2012.
- [DX14] Ciprian Dobre and Fatos Xhafa. Intelligent services for big data science. *Future generation computer systems*, 37:267–281, 2014.

- [ELSW06] Alexander M Ernst, Josef Lankes, Christian M Schweda, and André Wittenburg. Tool support for enterprise architecture management-strengths and weaknesses. In *2006 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06)*, pages 13–22. IEEE, 2006.
- [FDCD12] Danyel Fisher, Rob DeLine, Mary Czerwinski, and Steven Drucker. Interactions with big data analytics. *interactions*, 19(3):50–59, 2012.
- [GBMP13] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.
- [Gmb20] AKARA Solutions GmbH. Produkte, 2020. Zugriffsdatum: 24.11.2020. URL: <https://akara-solutions.de/produkte/>.
- [GMF⁺06] Shirley Gregor, Michael Martin, Walter Fernandez, Steven Stern, and Michael Vitale. The transformational dimension in the realization of business value from information technology. *The Journal of Strategic Information Systems*, 15(3):249–270, 2006.
- [GP11] Danny Greefhorst and Erik Proper. *Architecture principles: the cornerstones of enterprise architecture*. Springer Science & Business Media, 2011.
- [GR11] John Gantz and David Reinsel. Extracting value from chaos. *IDC iview*, 1142(2011):1–12, 2011.
- [Groat] The Open Group. Archimate® 3.1 specification. Zugriffsdatum: 05.10.2020. URL: <https://pubs.opengroup.org/architecture/archimate3-doc/toc.html>.
- [Grob] The Open Group. The archimate® enterprise architecture modeling language. Zugriffsdatum: 22.09.2020. URL: <https://www.opengroup.org/archimate-forum/archimate-overview>.
- [GSK15] J. Gao, S. Selle, and A. Koronios. Towards a process view on critical success factors in big data analytics projects. *Twenty-first Americas Conference on Information Systems, Puerto Rico, 2015*, 2015.
- [HPK11] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

- [HWCL14] Han Hu, Yonggang Wen, Tat-Seng Chua, and Xuelong Li. Toward scalable systems for big data analytics: A technology tutorial. *IEEE access*, 2:652–687, 2014.
- [ISO11] Iso/iec/ieee systems and software engineering – architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pages 1–46, 2011. doi:10.1109/IEEESTD.2011.6129467.
- [JfRfWA⁺17] Steven Ji-fan Ren, Samuel Fosso Wamba, Shahriar Akter, Rameshwar Dubey, and Stephen J Childe. Modelling quality dynamics, business value and firm performance in a big data analytics environment. *International Journal of Production Research*, 55(17):5011–5026, 2017.
- [KAE⁺09] Richard T Kouzes, Gordon A Anderson, Stephen T Elbert, Ian Gorton, and Deborah K Gracio. The changing paradigm of data-intensive computing. *Computer*, 42(1):26–34, 2009.
- [KAEM13] Stephen Kaisler, Frank Armour, J Alberto Espinosa, and William Money. Big data: Issues and challenges moving forward. In *2013 46th Hawaii International Conference on System Sciences*, pages 995–1004. IEEE, 2013.
- [Kal12] Ravi Kalakota. Big data infographic and gartner 2012 top 10 strategic tech trends, 2012.
- [KP15] Helena Kościelniak and Agnieszka Puto. Big data in decision making processes of enterprises. *Procedia Computer Science*, 65:1052–1058, 2015.
- [LAB⁺08] Anne Lapkin, Phillip Allega, Brian Burke, Betsy Burton, R Scott Bittler, Robert A Handler, Greta A James, Bruce Robertson, David Newman, Deborah Weiss, et al. Gartner clarifies the definition of the term ‘enterprise architecture’. *Gartner research*, 2008.
- [LGPA⁺13] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Jan Blom, Olivier Bor-net, Trinh Minh Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. From big smartphone data to worldwide research: The mobile data challenge. *Pervasive and Mobile Computing*, 9(6):752–771, 2013.
- [Man11] Lev Manovich. Trending: The promises and the challenges of big social data. *Debates in the digital humanities*, 2(1):460–475, 2011.

- [MCB⁺11] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, et al. *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute, 2011.
- [MG⁺11] Peter Mell, Tim Grance, et al. The nist definition of cloud computing. 2011.
- [MH13] ABM Moniruzzaman and Syed Akhter Hossain. Nosql database: New era of databases for big data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.
- [MJS13] Soumendra Mohanty, Madhu Jagadeesh, and Harsha Srivatsa. *Big data imperatives: Enterprise 'Big Data'warehouse, 'BI'implementations and analytics*. Apress, 2013.
- [MS13] Merja Mahrt and Michael Scharkow. The value of big data in digital media research. *Journal of Broadcasting & Electronic Media*, 57(1):20–33, 2013.
- [MU03] Günter Matthiessen and Michael Unterstein. *Relationale Datenbanken und SQL*. Pearson Deutschland GmbH, 2003.
- [ODS13] Aisling O'Driscoll, Jurate Daugelaite, and Roy D Sleator. 'big data', hadoop and cloud computing in genomics. *Journal of biomedical informatics*, 46(5):774–781, 2013.
- [OLPW⁺08] Martin Op't Land, Erik Proper, Maarten Waage, Jeroen Cloo, and Claudia Steghuis. *Enterprise architecture: creating value by informed governance*. Springer Science & Business Media, 2008.
- [Osv01] Gundars Osvalds. Definition of enterprise architecture-centric models for the systems engineers. *TASC Inc*, 2001.
- [Rag18] Elisabetta Raguseo. Big data technologies: An empirical investigation on their adoption, benefits and risks for companies. *International Journal of Information Management*, 38(1):187–195, 2018.
- [Ram19] Eike-Christian Ramcke. *Universelle Modellanalyse von Anwendungslandschaften basierend auf dem ArchiMate Standard*. PhD thesis, Hochschule für angewandte Wissenschaften Hamburg, 2019.

- [RMRESC⁺16] Lisbeth Rodríguez-Mazahua, Cristian-Aarón Rodríguez-Enríquez, José Luis Sánchez-Cervantes, Jair Cervantes, Jorge Luis García-Alcaraz, and Giner Alor-Hernández. A general perspective of big data: applications, tools, challenges and trends. *The Journal of Supercomputing*, 72(8):3073–3113, 2016.
- [Sch04] Jaap Schekkerman. *How to survive in the jungle of enterprise architecture frameworks: Creating or choosing an enterprise architecture framework*. Trafford Publishing, 2004.
- [SDC⁺16] Salman Salloum, Ruslan Dautov, Xiaojun Chen, Patrick Xiaogang Peng, and Joshua Zhexue Huang. Big data analytics on apache spark. *International Journal of Data Science and Analytics*, 1(3-4):145–164, 2016.
- [SF13] Pramod J Sadalage and Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2013.
- [SS17] Veda C Storey and Il-Yeol Song. Big data technologies and management: What conceptual modeling can do. *Data & Knowledge Engineering*, 108:50–67, 2017.
- [Sto12] Michael Stonebraker. Newsql: An alternative to nosql and old sql for new oltp apps. *Communications of the ACM*. Retrieved, pages 07–06, 2012.
- [Sut14] Shan Suthaharan. Big data classification: Problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review*, 41(4):70–73, 2014.
- [VBH15] Martin Vanauer, Carsten Böhle, and Bernd Hellingrath. Guiding the introduction of big data in organizations: A methodology with business- and data-driven ideation and enterprise architecture management-based implementation. In *2015 48th Hawaii International Conference on System Sciences*, pages 908–917. IEEE, 2015.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Bachelorarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Enterprise Architecture Management Metamodell zur Abbildung datenintensiver Prozesse

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____ 

Ort

Datum

Unterschrift im Original