Wintersemester 2019/2020

Visualisierungen für IT-Konsolidierungsprojekte

Thesis

zur Erlangung des Grades

Bachelor of Science

im Studiengang WirtschaftsNetze

an der Fakultät Wirtschaftsinformatik

der Hochschule Furtwangen University

vorgelegt von

Katharina Schemel

Referenten:	Prof. Dr. Marianne Andres Martin Jelen
Eingereicht am	13. Februar 2020

Eidesstattliche Erklärung

Ich Katharina Schemel erklären hiermit an Eides statt, dass ich die vorliegende Bachelorthesis selbständig und ohne unzulässige fremde Hilfe angefertigt habe.

Die verwendeten Quellen sind vollständig zitiert.

Furtwangen, den 13. Februar 2020

_____

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

In der Realität ist es oft so, dass innerhalb eines Unternehmens viele verschiedene IT-Systeme verwendet werden, welche schnell schwierig zu überblicken sind. Diese Systeme beinhalten Programme, welche auch gepflegt werden müssen. Dies resultiert beispielsweise daraus, dass diese immer auf dem aktuellen Stand sein sollten, um Sicherheitslücken zu vermeiden. In großen Unternehmen hat man kaum einen Überblick, welche IT-Systeme in den einzelnen Abteilungen verwendet werden. Hier empfiehlt sich eine Konsolidierung vorzunehmen. Das bedeutet, dass die IT-Systeme sozusagen verschlankt werden.

Ein Aspekt der Konsolidierung ist die Ineffizienz von IT-Systemen zu eliminieren. Oft werden verschiedene Systeme verwendet, welche im Prinzip, das exakt gleiche tun. Daher wäre es von Vorteil eine einheitliche Struktur festzulegen, um den Bestand der Software zu vereinfachen.

Ein weiterer Aspekt bei der Konsolidierung ist die Entscheidung, welche Komponenten konsolidiert werden sollten. Zu beachten ist jedoch, dass durch die Konsolidierung nie eine konkrete Entscheidung gefällt wird welche Komponenten konsolidiert werden, sondern diese lediglich eine Entscheidungsgrundlage darstellt. Der Anstoß für eine IT-Konsolidierung kann aus aktuellen Anlässen erfolgen, strategischen Nutzen für das Unternehmen bieten oder einfach von der Unternehmensführung gewünscht sein.

All diese Gründe führen zu der Konsolidierung, um beispielsweise den Geschäftswert bzw. die Betriebskosten des Unternehmens zu optimieren oder zur Zuordnung von Geschäftsprozessen, welche zu Outsourcing führen können. Somit ist deutlich zu erkennen, dass die Konsolidierung ein wichtiger Aspekt in Unternehmen darstellt.

Das Problem bei der Konsolidierung ist jedoch, die einzelnen Komponenten zu identifizieren. Eine Datenbeschaffung ist recht zeitaufwendig und kann fast nie vollständig erfolgen. Hieraus entsteht meist ein Vorgehen auf Basis von Teilbeständen, was letztlich nicht dem Ideal entspricht.

Die Komponente Visualisierung wird sowohl für die Präsentation der Analyseergebnisse wie auch als Werkzeug für die Analyse eingesetzt. Um die Analyse zu verbessern ist es vorstellbar Graphdatenbanken in diesen Prozess mit einzubinden, da ein Graph schnell einen Überblick über Beziehungen und Zusammenhänge der einzelnen

Komponenten vermittelt. Daraus kann eine Entscheidungsgrundlage gebildet werden, welche Komponenten entsprechend eliminiert oder verändert werden sollten.

1.1 Motivation

Die Thematik der IT-Konsolidierung ist noch recht jung und sehr aktuell, beispielsweise hat die IT-Konsolidierung des Bundes erst 2015 begonnen. Daher ist es auch sehr spannend diese Thematik weiter voran zu bringen. Auslöser für eine IT-Konsolidierung kann z. B. eine vorherige Fusion darstellen. Laut (Bundesministeriums des Innern, für Bau und Heimat, 2020) sind die folgenden Punkte Ziele einer IT-Konsolidierung:

- Informationssicherheit trotz steigender Komplexität zu gewährleisten
- die eigene IT soll zu jederzeit Souverän und beherrscht sein
- flexibel auf innovative technologische Trends reagieren zu können
- ein zukunftsfähiger Betrieb, welcher leistungsfähig und stabil ist
- für das IT-Fachpersonal ein attraktiver Arbeitgeber bleiben

IT-Konsolidierung: Eine einfache Definition

> *„Konsolidierung bedeutet einfach übersetzt „Zusammenführung". Im Bereich der IT sollen also Infrastrukturen, Datenbestände und Anwendungen zusammengeführt und idealerweise auch vereinheitlicht werden. Dadurch werden Kosten gespart, Abläufe vereinfacht sowie beschleunigt und die IT-Qualität erhöht sich insgesamt."* (Klein, 2017)

Diese Definition zeigt, das IT-Konsolidierung sehr wichtig ist, da neben einer Kosteneinsparung auch die Qualität der IT erhöht werden soll, was insgesamt eine Bereicherung für die heutige Zeit darstellt, in welcher die Themen Digitalisierung und Industrie 4.0 immer allgegenwärtiger werden. Auch die Zusammenführung von Infrastrukturen, Datenbeständen und Anwendungen erleichtern das tagtägliche Arbeiten, sowie die Pflege der Software erheblich. Die Konsolidierung wird im Enterprise Architecture Management (EAM) durchgeführt, da diese einen konzeptionellen und organisatorischen Rahmen bietet, damit die Architektur zielgerichtet aufgebaut und erweitert werden kann. Zudem kann die Architektur Zusammenhänge z. B. mittels Diagramme sehr gut darstellen. Besonders bei komplexen Sachverhalten ist diese Weise deutlich effizienter.

Die Thematik der Visualisierung im Kontext IT-Konsolidierung ist ein wichtiger Bestandteil dieser Arbeit, da eine Präsentation der Analyseergebnisse mittels Excel-Tabellen, wie es häufig von IT-Beratern praktiziert wird, nur einen geringen Mehrwert bietet.

1.2 Ziel der Arbeit

Das Ziel der Arbeit besteht sozusagen in der „Grundsteinlegung“ auf dem Gebiet der Visualisierung von IT-Konsolidierungsprojekten mit Anbindung an eine Graphdatenbank für die Firma ISB AG. Die Arbeit gliedert sich somit in einen theoretischen wie auch einen praktischen Teil. Der theoretische Teil umfasst die Einarbeitung auf den Themengebieten EAM, Graphdatenbanken und Visualisierung mittels D3JS. Neben der Einarbeitung stellt sich die zentrale Frage, wie Graphdatenbanken und die Visualisierung auf dem Gebiet der IT-Konsolidierungsprojekte zusammen zu bringen sind. Hierfür sollten prototypische Visualisierungskonzepte erarbeitet werden, unter Berücksichtigung von typischen Visualisierungskonzepten aus dem Bereich des EAM.

Es sei jedoch zu beachten, dass nicht jeder Datensatz, aufgrund seiner spezifischen Eigenschaften, zu jedem Visualisierungskonzept passt. Aus diesem Grund sollten Datenbestände zunächst analysiert werden und dementsprechend eine geeignete Visualisierung aufgrund der Daten gewählt werden. Bezüglich einer Graphdatenbank folgt aus einer derartigen Analyse der Daten in der Regel ein Datenmodell, welches im Zuge dieser Arbeit für eine fiktive IT-Landschaft prototypisch zu erstellen war. Dieses wiederum bildete die Grundlage für die Erstellung der Visualisierungskonzepte.

Die Grundlagenlegung für den praktischen Teil resultierte wiederum in der Erweiterung des theoretischen Teils. Es ging im Zuge dessen darum ein Konzept für eine Infrastruktur für den Betrieb des im praktischen Teil entwickelten Prototyps zu generieren. Weiterhin musste ein Konzept für die Entwicklungsumgebung geschaffen und entsprechend umgesetzt werden, um eine nachhaltige und einfache Organisation des zu erstellenden Softwareprojekts zu gewährleisten. Folglich umfasst das Ziel des praktischen Teils die Entwicklung eines Prototyps, welcher diverse Kernfunktionalitäten exemplarisch implementiert. Diese sind unteranderem die Anbindung an eine Graphdatenbank sowie die Visualisierung der angefragten Daten mittels der JavaScript-Bibliothek D3JS.

2 Grundlagen

2.1 Enterprise Architecture Management

Die folgenden Definitionen von Enterprise Architecture und Enterprise Architecture Management zeigen die konkreten Abgrenzungen voneinander auf. In der Praxis werden diese Begriffe oft als Synonym verwendet, jedoch gibt es gravierende Unterscheidungen zwischen diesen beiden Begriffen.

Definition Unternehmensarchitektur (EA):

„Eine Unternehmensarchitektur (Enterprise Architecture) schafft eine gesamthafte Sicht auf das Unternehmen. Sie legt die wesentlichen fachlichen und IT-Strukturen fest und verknüpft sie miteinander. Auf dieser Basis lassen sich das Business und die IT und ihre Zusammenhänge beschreiben. Eine gemeinsame Sprachbasis, „eine Brücke" zwischen Business und IT, wird geschaffen. So kann die strategische Weiterentwicklung von Business und IT aktiv gesteuert werden." (Hanschke, 2013)

Definition EAM:

„EAM ist ein systematischer und ganzheitlicher Ansatz für das Verstehen, Kommunizieren, Gestalten und Planen der fachlichen und technischen Strukturen im Unternehmen. Es hilft dabei, die Komplexität der IT-Landschaft zu beherrschen und die IT-Landschaft strategisch und businessorientiert weiterzuentwickeln. EAM ist ein wesentlicher Bestandteil des strategischen IT-Managements und beinhaltet alle Prozesse für die Dokumentation, Analyse, Qualitätssicherung, Planung und Steuerung der Weiterentwicklung der IT-Landschaft und der Geschäftsarchitektur." (Hanschke, 2013)

Einfach gesagt, fungiert die Enterprise Architecture als eine Art Vermittlerrolle zwischen der IT und den entsprechenden Fachbereichen eines Unternehmens, so (BITKOM, 2011). Man könnte es auch bildlich als eine Brücke beschrieben, welche diese Bereiche verbindet.

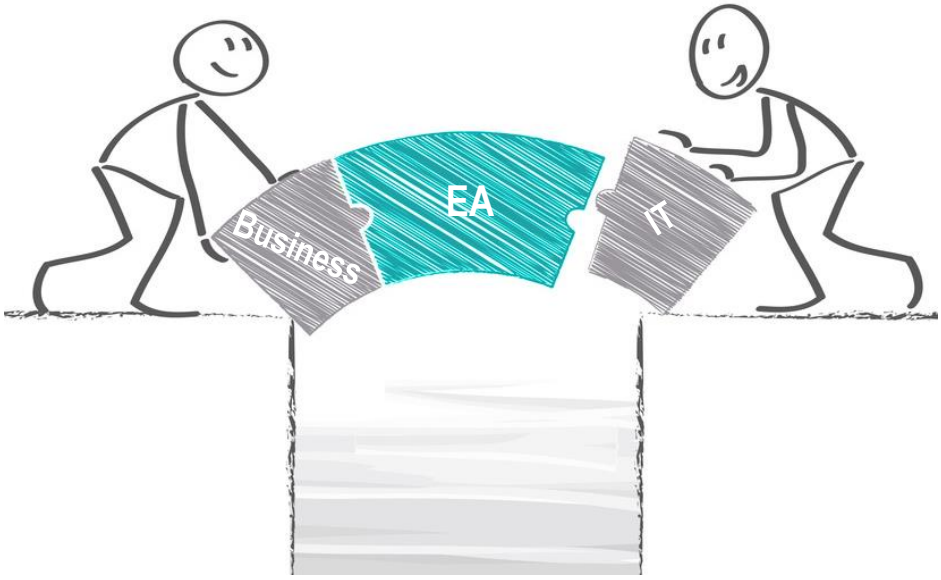

Abbildung 1: Brücke zwischen Business und IT (Business-IT)

Durch diese Verbindung wird ein Rahmen für den Ausbau der IT-Landschaft zur Verfügung gestellt. Dieser Rahmen umfasst laut (BITKOM, 2011) strategische, konzeptionelle und organisatorische Aspekte. Weiter führt (BITKOM, 2011) aus, dass besonders die Methoden der Umsetzung und die Überprüfung im Hinblick einer Kosten-Nutzen-Betrachtung im Fokus stehen.

Das Enterprise Architecture Management wiederum „*umfasst die Aufgaben zur Erstellung, Pflege und Umsetzung einer EA*“ (BITKOM, 2011). Man kann also sagen, dass das EAM ein strukturierter Ansatz für die Erstellung, Verwaltung und Nutzung der von EA bereitgestellten Modelle ist.

Je größer ein Unternehmen wird, desto bedeutender ist die Rolle des EAM. Durch neue Technologien und Schnittstellen steigt die Komplexität und das Bewältigen der IT-Landschaft wird immer schwieriger. Durch fehlendes EAM können Redundanzen und inkonsistente Daten auftreten. Diese Problematik ist unter anderem die Basis für die IT-Konsolidierung. Aber auch mit EAM kann über einen längeren Zeitraum eine IT-Konsolidierung notwendig werden.

Durch das Bereitstellen diverser Hilfsmittel durch EAM, kann die Kontrolle über die IT-Landschaft wiederhergestellt werden und sie kann strategisch und businessorientiert weiterentwickelt werden.

Zudem wird EAM oft eingesetzt, da es Transparenz schafft. Diese ist maßgeblich für die Bewertung der IST-Situation und zeigt bei der IT-Landschaft schon nach kurzer Zeit, in welchem Bereich sich Kosten einsparen lassen.

Laut (Hanschke, 2013) kann in Problemfällen durch entsprechende Visualisierungen schneller Klarheit verschafft werden, z. B. welche Geschäftsprozesse von einem Ausfall des IT-Systems betroffen sind, wo die entsprechenden Verantwortlichkeiten liegen oder welche Abhängigkeiten zwischen den IT-Systemen bestehen.

Um die Leistungsfähigkeit der IT-Leistung einordnen zu können, werden in der Praxis oft relative Bewertungen wie z. B. sehr hoch, hoch, mittel und gering verwendet. Diese Bewertung ist einfach und verständlich, damit sie von Fachbereichen, Anwendern und Managern angewendet werden kann. Die Fachbereiche und die Fachfremden Personen sollten jeweils eine solche Bewertung vornehmen, um festzustellen in wie weit sich das Eigen- und Fremdbild unterscheidet und anschließend sollte entsprechend vorgegangen werden, um diese beiden Ansichten anzugleichen, führt (Hanschke, 2013) weiter aus.

2.2 Visualisierungskonzepte

Farben und Muster sprechen den Menschen schneller an als eine einfache Tabelle mit Informationen. Daher ist eine Visualisierung von Daten wichtig, um das Interesse schneller auf relevante Aspekte zu lenken. Bei der Darstellung von Inhalten ist es wichtig sich vorab zu überlegen, welcher Sachverhalt näher dargestellt werden soll und entsprechend können hierfür unterschiedliche Visualisierungsarten mit passenden Eigenschaften herangezogen werden.

Grundsätzlich sei an dieser Stelle darauf verwiesen, dass nicht jede Visualisierungsart für die Darstellung eines beliebigen Sachverhalts geeignet ist. Beispielsweise kann ein Ampel Diagramm sehr gut verwendet werden, um einen Fortschritt in einem Prozess oder die Kritikalität einer Eigenschaft darzustellen. Es ist jedoch nicht geeignet komplexere Beziehungen zwischen Personen, Prozessen oder Softwareprodukten darzustellen.

Auch für den Bereich des EAM gibt es mehrere gängige Visualisierungsarten, welche laut (Hanschke, 2013) die Aussagekraft der Informationen unterstreicht. Auch auf diesem Gebiet gestaltet sich der Einsatzbereich solcher Grafikkonzepte analog zum Einsatzberiech der Ampel Grafik. Sie sind jeweils Anhand ihrer Eigenschaften für die Darstellung gewisser Sachverhaltet besser geeignet und für andere schlechter. Beispiele für Visualisierungsarten aus dem Bereich des EAMs seien gegeben durch folgende Grafikkonzepte:

- Bebauungsplan-Grafik
- Cluster-Grafik
- Portfolio-Grafik

Die drei genannten Grafikkonzepte wurden exemplarisch aus dem Best-Practice-Visualisierungskatalog gewählt. Diese zeigen durch ihre unterschiedlichen Ausprägungen die verschiedenen Möglichkeiten, welche eine Visualisierung im Bereich des EAM leisten kann. Diese drei Grafikkonzepte werden folgend näher erläutert und zur Veranschaulichung nach Vorlage von (Hanschke, 2013, 2016) nachgestellt.

- Bebauungsplan-Grafik

  Diese Grafik stellt die Zusammenhänge zwischen Elementen der Unternehmensarchitektur in Form einer Matrix dar. Somit können u.a. Informationssysteme zu Geschäftsprozessen und Geschäftseinheiten in Beziehung gesetzt werden. Durch die flexible Struktur können viele Fragestellungen beantwortet werden. Der Einsatz von Farben und diversen Linientypen unterstreicht diese Visualisierung nochmals. Die Bebauungsplan-Grafik unterteilt sich in drei Unterprunkte. Für das EAM ist besonders die Ausprägung der „typischen“ Bebauungsplan-Grafik interessant, da diese eine verbreitete Form zur IT-Unterstützung des Unternehmens ist. In einem Bebauungsplan der IT-Branche werden laut (BITKOM, 2011) gegenwärtige und zukünftige Infrastrukturen, als auch Anwendungssoftware definiert. Diese sollen die Geschäftsprozesse eines Unternehmens unterstützen.

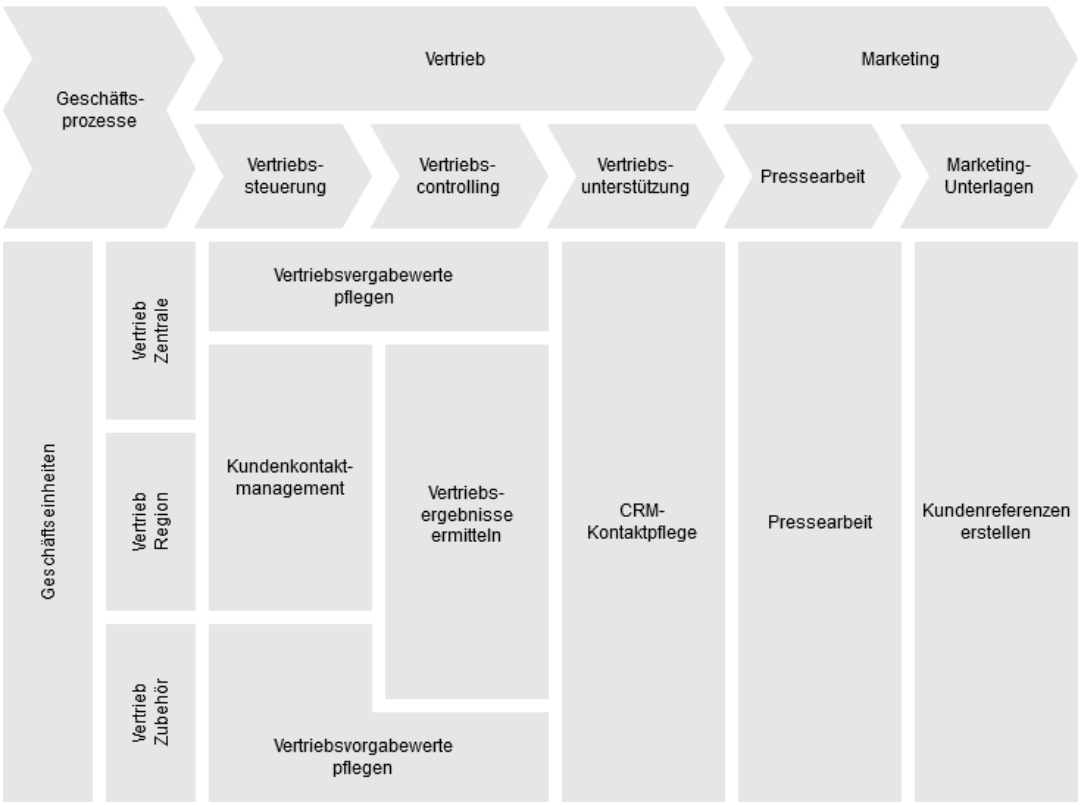

Abbildung 2: Bebauungsplan Grafik

Diese Abbildung zeigt die Geschäftsprozesse auf der X-Achse und die Geschäftseinheiten des Vertriebs auf der Y-Achse. Die einzelnen Bebauungselemente in mitten dieser Grafik geben einen guten Überblick, welche Geschäftseinheiten für welche Geschäftsprozesse zuständig sind.

Diese Art der Grafik ist jedoch nicht zwingend kompatibel zu größeren Datenmengen mit einer netzwerkartigen Struktur, da Abhängigkeiten unter den Daten deutlich schwieriger dargestellt werden können. Dies resultiert aus der blockartigen Anordnung, welche ab einer gewissen Datenmenge eine optische Clusterung nur bedingt zulässt.

- Cluster-Grafik

  Das Konzept der Cluster-Grafik beschreibt das Aufteilen und Gruppieren von Bebauungselementen, anhand von Eigenschaften, Beziehungen oder definierten Kriterien. Bekannte Ausprägungen der Cluster-Grafik sind z. B. das Fachliche Domänenmodell, welches die Ausprägung einer Prozesslandkarte oder eines funktionalen Referenzmodells hat. Im Fachlichen Domänenmodell werden die Kernstrukturen der Geschäftsarchitektur festgelegt, daher gibt es laut (Hanschke, 2013) innerhalb eines Unternehmens auch nur ein Fachliches Domänenmodell.

  Grafik

Abbildung 3: Funktionales Referenzmodell

Die Abbildung 3 zeigt ein funktionales Referenzmodell. Dieses Eignet sich bis zu einem gewissen Grad, Abhängigkeiten zwischen Bebauungselementen darzustellen. Dies wird in diesem Fall auf Basis einer verschachtelten Darstellung realisiert. Somit eignet sich diese Darstellungsart, um die Zusammengehörigkeit zwischen Elementen der IT-Landschaft eines Unternehmens darzustellen. Die Grenzen werden jedoch erreicht, sobald Elemente in mehreren Gruppierungen zugehörig sind, da dies maximal durch eine redundante Aufführung kompensiert werden kann. Dies kann jedoch bei größeren Darstellungen wiederum die Lesbarkeit bzw. Interpretierbarkeit der Grafik beeinträchtigen. Es kann somit auch nicht auf einfachem Wege dargestellt werden wie stark ein Softwareprodukt in ein Unternehmen integriert ist.

- Portfolio-Grafik

  Bei dieser Grafik können besonders gut sogenannte Wertigkeiten von Bebauungselementen oder auch Strategien für Bebauungselemente visualisiert werden. Ebenso eignet sie sich für die Abbildung von Nutzenpotenzialen oder Risiken. Hierbei kann der Ist-Zustand oder auch der Soll-Zustand bzw. deren Differenz dargestellt werden. In einer Portfolio Grafik können maximal fünf verschiedene Kriterien abgebildet werden. Diese sind jeweils die Achsen, sowie die Größe, Farbe und der Kantentyp der Füllelemente.

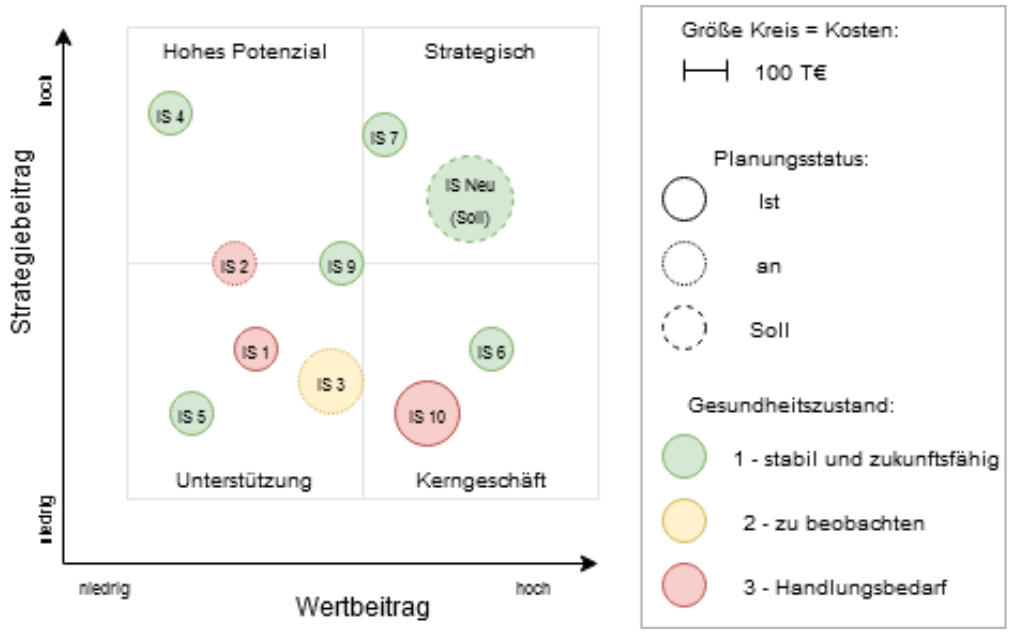

Abbildung 4: Portfolio-Grafik

Durch diese Grafik kann schnell ein Überblick anhand der Positionierung verschafft werden. Neben der Position, gibt auch die Größe der einzelnen Elemente Aufschluss über beispielsweise Kosten. Diese Grafik eignet sich besonders gut, um messbare Eigenschaften oder eine kategorische Einordnung von Elementen darzustellen. Abhängigkeiten zwischen Elementen können maximal über die kategorische Einordung realisiert werden.

2.3 Funktionsweise einer Graph-Datenbank

Jeder ist bereits mit einem Graphen in Berührung gekommen. Sei es bei der Erstellung eines Mindmaps oder beim Skizzieren von Symbolen und Linien auf einer Tafel. Graphen sind sehr einfach, verständlich und vielseitig einsetzbar, nach (Hunger, 2014). Eine der bekanntesten Graph Datenbanken ist Neo4j. Der Name Neo4j leitet sich aus *„Network Engine for Objects“* ab. Die Zahl „4“ vermittelt eine Aussage über die Versionsnummer und das „j“ steht dafür, dass die frühere Java-API entsprechend weiterentwickelt wurde. Zu Beginn war die Bedeutung von „4j“ „für Java“, führt (Trelle, 2017) aus. Mittlerweile kann, neben Java, auch in der Programmiersprache Scala entwickelt werden. Daher ist die Bedeutung „for Java“ nicht mehr ganz aktuell.

Eine Graph-Datenbank gehört zu der Gruppe der NoSQL Datenbanken. Dies bedeutet, dass die Datenbank in der Regel einen nicht relationalen Ansatz verfolgt. Jedoch wird die Verwendung von SQL nicht gänzlich ausgeschlossen, da NoSQL für „Not only SQL“ steht.

Eine Relationale Datenbank verwendet Tabellen, welche Spalten und Zeilen für die Speicherung der Daten nutzt. Die NoSQL Datenbank hingegen nutzt für die Organisation der Daten beispielsweise Attribut-Wert-Paare (Properties), Objekte, Dokumente oder Listen und Reihen für die Organisation der Daten. Ein großer Vorteil von NoSQL Datenbanken ist, dass sie dort ansetzen, wo SQL-basierte relationale Datenbanken an ihre Grenzen stoßen. NoSQL Systeme eignen sich besonders gut für große, exponentiell wachsende Datenmengen, um diese performant zu verarbeiten, wie beispielsweise im Bereich von Big Data. Dies resultiert aus der einfachen Unterstützung der Datenbankfragmentierung, welche durch den Verzicht auf ein Datenbankschema und auf die referenzielle Integrität ermöglicht wird. Im speziellen besagt die referenzielle Integrität, dass Datensätze nur auf bestehende Datensätze verweisen dürfen. Entsprechend kann auch nur ein Datensatz gelöscht werden, wenn auf diesen kein anderer Datensatz verweist.

Graph Datenbanken werden nicht tabellarisch geführt, sondern in Form von Labeln organisiert. Diese sind zwar vergleichbar mit den Tabellen der relationalen Datenbanken, haben jedoch den Unterschied, dass diese ohne eine feste Vorgabe der enthaltenen Attribute auskommt. Die Speicherung der Daten erfolgt somit Schemafrei. Dies erlaubt zusätzliche Informationen wie mangelnde Attribute zu einem späteren Zeitpunkt problemlos nachzutragen. Ein implizietes sturkturierndes Schema sei dennoch durch das Graphenmodell innerhalb einer Graphdatenbank gegeben. Das Graphenmodell kann als Äquivalent zum Relationenmodell der Relationalen Datenbanken interpretiert werden. Es resultiert analog zum Relationenmodell aus dem Entitäten-Beziehungsmodell. Dieser Zusammenhang sei mit Abbildung 5 nach (Meier et al., 2016) näher verdeutlicht.

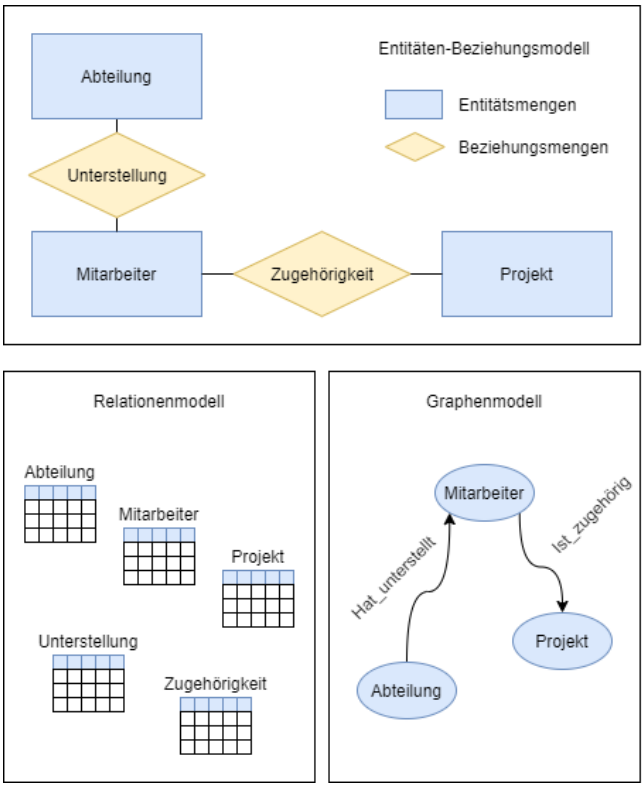

Abbildung 5:Zusammenhang Entitäten-Beziehungsmodell, Relationenmodell, Graphenmodell

Laut (Maier et al., 2016) werden die zu speichernden Daten mithilfe von so genannten Knoten und Kanten dargestellt. Hier werden Objekte in Knoten und Beziehungen zwischen Knoten in Kanten abgebildet. In einem Knoten ist die Beziehung nicht in ihrer Art oder Anzahl beschränkt, ergänzt (Matzer, 2019).

In der Literatur werden Knoten häufig auch als Vertex (V) und Kanten als Edges (E) benannt. Dies sei an dieser Stelle bezüglich der Vollständigkeit erwähnt. Folgend wird diese Terminologie jedoch nicht weiterverwendet.

Die Kante stellt eine Verbindung zwischen den Knoten dar. Diese Verbindung kann als eine Linie oder als ein Pfeil visualisiert werden. Eine Kante ist gerichtet, das bedeutet, dass sie einen Start- sowie einen Endpunkt besitzt. Die Beziehungen zwischen den Knoten können Eigenschaften besitzen. Diese Daten können analog zu den Attributen der Knoten abgefragt werden. Nach (Luber et al., 2017) verzichtet eine Graphdatenbank auf verschachtelte Beziehung, was die hohe Performance für die Speicherung, sowie Abfrage der Informationen erklärt.

Die Information, welche Beziehungen zwischen Daten vorliegen, ist unter gewissen Umständen besonders förderlich. Diese Beziehungen kann ein Graph ausgesprochen gut darstellen. Hieraus ergibt sich der hohe Nutzen der Verwendung einer Graph Datenbank für Daten, welche eine netzwerkartige Struktur aufweisen.

Eine Besonderheit der Graphdatenbanken ist nach (Maier et al., 2016), die Eigenschaft der indexfreien Nachbarschaft. Das Datenbanksystem kann somit die direkten Nachbarn eines Knotens ermitteln, ohne sämtliche Kanten berücksichtigen zu müssen. Dies ist beispielsweise in relatioalen Datenbanksystemen nötig, welche sogenannte Beziehungs- oder Verknüpfungstabellen verwenden. Aus diesem Sachverhalt ergibt sich, dass die Abfrage von Beziehungen auf Knoten unabhängig von der gespeicherten Datenmenge sind und somit eine konstante Geschwindigkeit besitzen.

Laut (Matzer, 2019) nutzen bereits große Unternehmen wie z. B. Google, Facebook, Microsoft und LinkedIn die Graph Datenbank, ebenso haben auch Unternehmen wie die NASA oder die Schweizer Bank UBS die Graph Datenbank für sich entdeckt.

Die gängigsten Abfragesprachen für Graph Datenbanken sind Apache TinkerPop Gremlin, SPARQL und Cypher. Da, in dieser Arbeit mit der Graph Datenbank Neo4j gearbeitet wird, betrachtet diese Arbeit folgend nur die Abfragesprache Cypher näher. Cypher wurde von Neo4j entwickelt und erfüllt somit die besten Voraussetzungen was Kompatibilität angeht, nach (Matzer, 2019). Zudem ist Cypher sehr verbreitet und in seiner Struktur der Abfragesprache SQL von Relationalen Datenbanken recht ähnlich. Durch diese ähnliche Struktur ist eine Umstellung von SQL auf Cypher nicht zu zeitaufwendig und komplex. In Kapitel 2.4 wird auf die Abfragesprache Cypher näher eingegangen.

2.4 Cypher

Cypher gehört zu der Gruppe der „deklarativen Abfragesprachen“. Die bekannteste Abfragesprache dieser Gruppe ist SQL. Der Fokus der deklarativen Abfragesprachen ist die Beschreibung eines Problems. Gegenüber den deklarativen Abfragesprachen steht die Gruppe der „imperativen Sprachen“, wie beispielsweise Java oder C++. Nach (Böhm, 2005) ist die imperative Programmierung ein Programmierstil, nach welchem „ein Programm aus einer Folge von Anweisungen besteht, die vorgeben, in welcher Reihenfolge was vom Computer getan werden soll“.

Wird folgend Cypher als problembeschreibende Sprache in Bezug auf Graphen betrachtet, kann von der Beschreibung von Mustern innerhalb von Graphen gesprochen werden. An folgendem Beispiel wird dieser Sachverhalt verdeutlicht. Es wird ein Graph angenommen, bestehend aus zwei Klassen von Knoten und einer gerichteten Beziehung zwischen den Knoten. Exemplarisch seien dies für die Knoten „Filme“ und „Schauspieler und für die Beziehung „spielt_in“. Entsprechend beschreibt der exemplarische Graph, welcher Schauspieler in welchem Film spielt. Die Beschreibung des Musters des dargelegten Graphen mittels Cypher stellt sich wie folgt dar.

```
Match (s:Schauspieler) – [spielt_in] -> (f:Film) Return s, f
```

Die entsprechende Rückgabe sind alle Knoten und deren Verbindungen untereinander, wie in Abbildung 6 dargestellt. Filme, zu denen keine Schauspieler in der Datenbank hinterlegt sind, werden nicht ausgegeben, da sie dem abgefragten Muster nicht entsprechen.

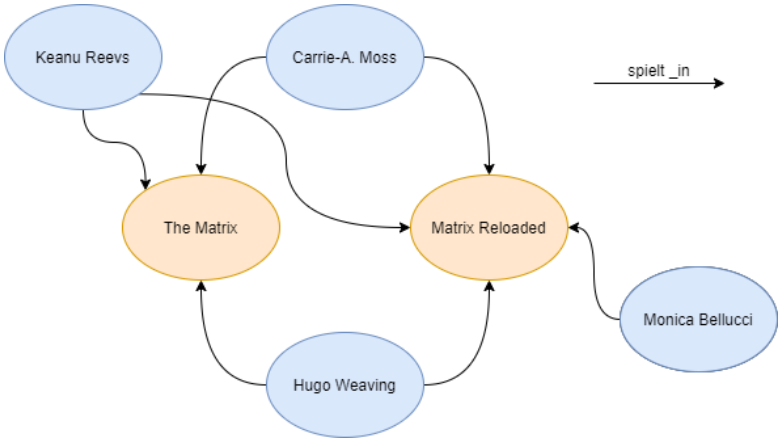

Abbildung 6: Ergebniss Relationenabfrage

Analog zu SQL bietet Cypher die Option, anhand von Attributen der Knoten und Kanten mittels einer Where-Bedingung das Muster zu Filtern. Folgend sei dies mittels Cypher dargestellt.

```
Match (s:Schauspieler) – [spielt_in] -> (f:Film)
Where (s.name = „Carrie-A. Moss)
OR (s.name = „Monica Bellucci“)
Return s, f
```

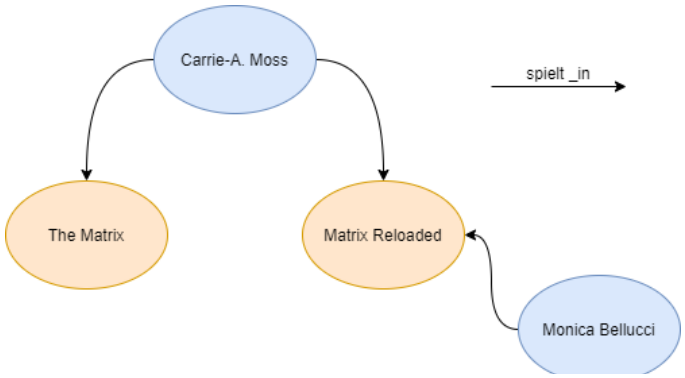

Abbildung 7: Ergebniss Relationenabfrage mit Einschränkungen auf Attributwerte

2.5 Node.js

Node.js ist eine plattformübergreifende Laufzeitumgebung basierend auf JavaScript. Verwaltet wird das Open-Source-Projekt NodeJS durch die OpenJS Foundation mit Unterstützung der Linux Foundation. Die ursprüngliche Hauptaufgabe von NodeJS besteht in der Ausführung von JavaScript-Code außerhalb eines Browsers. Es wird somit möglich serverseitig dynamische Webseiteninhalte zu erstellen, bevor die Seite an den Benutzer übertragen wird. Entsprechend wird deutlich, dass NodeJS primär für die Entwicklung hoch performanter Webserver verwendet wird. Nativ wird von NodeJS nur JavaScript unterstützt. Es gibt jedoch Compiler, die es ermöglichen andere Sprachen, wie zum Beispiel CoffeeScript oder TypeScript in JavaScript zu übersetzen und somit für NodeJS nutzbar zu machen.
Auf der funktionalen Ebene ist NodeJS bezüglich der dynamischen Erstellung von Webseiteninhalten am ehesten mit PHP zu vergleichen, welches dies analog zu NodeJS ebenso unterstützt. Auf der technischen Ebene unterscheidet sich NodeJS jedoch deutlich von PHP. NodeJS führt Funktionen in der Regel parallel aus und verwendet sogenannte „Callbacks“ um die Fertigstellung oder das Scheitern einer Funktion zu signalisieren. PHP hingegen blockiert die meisten Funktionen bis zur Fertigstellung der vorherigen.
Neben der beschriebenen Hauptaufgabe wird NodeJS jedoch auch sukzessive als Werkzeug für diverse andere Aufgaben eingesetzt. In dieser Arbeit wurde NodeJS beispielsweise indirekt als Bestandteil der Entwicklungs- und Deploymentumgebung verwendet. Dies wird ermöglicht, durch den Node Package Manager (npm), welcher ein integrierter Bestandteil von NodeJS ist. Der npm ist ein Paketmanager für die Laufzeitumgebung von NodeJS, was letztlich als freies Repository für Paketerweiterungen für NodeJS interpretiert werden kann. Am Beispiel dieser Arbeit wurde eine Reihe von Paketerweiterungen wie beispielsweise „webpack“ oder „babel“ für die Entwicklung und das Deployment eingesetzt. Diese Erweiterungen reichen von Compileren bis hin zu Webservern für die lokale Entwicklung. Auf die im Rahmen dieser Arbeit eingesetzten Paketerweiterungen wird in Kapitel 4 spezifischer eingegangen.

2.6 Webserver

Ein Webserver ist schlicht eine Software, welche serverseitig installiert wird. Die Hauptfunktion eines Webservers ist nach (Killelea, 2002) das Speichern, Verarbeiten und Übermitteln von Webseiten an Clients. Er operiert somit im sogenannten „Client-Server Modell“, welches in Abbildung 8 näher veranschaulicht ist. Ein Webserver kann im Allgemeinen eine oder mehrere Webseiten enthalten. Bei den bereitgestellten Seiten handelt es sich am häufigsten um HTML-Dokumente, die neben dem Textinhalt auch Bilder, Stylesheets und Skripte enthalten können.
Für Clientanfragen und Serverantworten ist das Hypertext Transfer Protocol (HTTP) von zentraler Bedeutung. Der Client stellt eine HTTP-Anfrage mit einem Uniform Resource Locator (URL), durch welchen die angeforderte Ressource serverseitig identifiziert werden kann. Anschließend antwortet der Webserver nach der Anfragenverarbeitung dem Client mit einer entsprechenden Nachricht. Diese enthält wiederum Informationen zum Status der Verarbeitung (erfolgreich oder gescheitert) und gegebenenfalls den angeforderten Inhalt beziehungsweise die angeforderte Ressource.

Während die Hauptfunktion darin besteht, Inhalte bereitzustellen, umfasst eine vollständige Implementierung von HTTP auch Möglichkeiten zum Empfangen von Inhalten. Diese Funktionalität wird zum Senden von Webformularen verwendet bis einschließlich des Hochladens von Dateien.

Viele generische Webserver unterstützen wie schon in Kapitel 2.4 aufgegriffen das serverseitige Skripting mit beispielsweise PHP (Hypertext Preprocessor) oder anderen Skriptsprachen. Dies bedeutet letztlich, dass das Verhalten des Webservers mittels separater Skripte erweitert beziehungsweise definiert werden kann, während die tatsächliche Serversoftware komplett unberührt bleibt. Die dynamische Generierung von Webseiteninhalten wird überwiegend zum Abrufen oder Ändern von Informationen aus Datenbanken verwendet.

Die aktuell am weitesten verbreiteten Webserver nach (w3techs.com, 2020) sind der Apache HTTP Server und der Nginx Webserver.

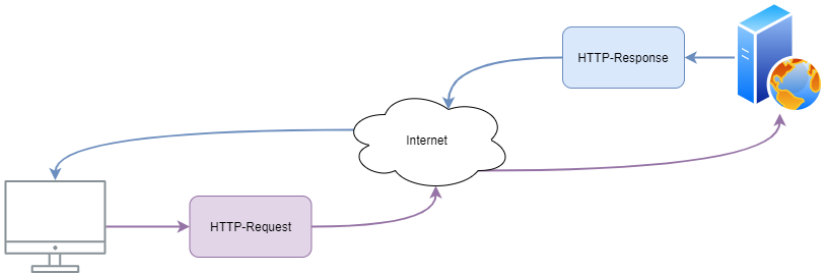

Abbildung 8: Client-Server Modell

2.7 Webtechnologien zur Visualisierung (Überdenken ob passend)

Der Sinn einer Visualisierung ist, dass diese auch gesehen wird und der schnellste und einfachste Weg eine Visualisierung für mehrere Menschen zugänglich zu machen ist über eine Webseite. Ein Webbrowser hat laut (Murray, 2017) den Vorteil, dass er nicht an ein Betriebssystem wie beispielsweise Windows, Mac OS, Linux, Android oder iOS gebunden ist, sondern für die Anwendung nur einen Internetzugang und einen Browser benötigt. Ein weiterer Vorteil ist, dass diese Variante kaum Lizenz- und Schulungskosten mit sich bringt. Jedoch muss hierbei auch die Benutzerfreundlichkeit im Fokus liegen, da der Anwender nicht zwingend technisch begabt ist, sollte die Webansicht einfach, verständlich und intuitiv sein. Laut (Wirtschaftslexikon Onpulson) müssen, um die Benutzerfreundlichkeit und die Erwartungen der Zielgruppe aufeinander abzustimmen, die Usability-Kriterien Effektivität, Effizienz und Zufriedenheit mit dem Content, Design und der Struktur der Webseite harmonieren. Mit Effektivität ist hier gemeint, dass der Anwender auf dem richtigen Weg ist, um etwas zu finden, das bedeutet, dass eine Seite so aufgebaut sein muss, dass man sich einfach zurechtfindet und auch zu jeder Zeit weiß wo man sich aktuell befindet. Effizient steht hier für die Schnelligkeit. Der Nutzer möchte schnell sein gewünschtes Ziel erreichen können. Und die Zufriedenheit sagt aus, ob sich der Anwender in einem für ihn ästhetischem Umfeld befindet also wie ihn die Seite optisch anspricht.

Dazu gibt es auch kommerzielle Tools für eine Visualisierung von Graphen. Diese sind z. B. KeyLines und Linkurious. Diese bieten zwar einige Analysemöglichkeiten, jedoch werden diese nicht weiter in Betracht gezogen, da das Unternehmen ISB AG eine kostengünstige Lösung anstrebt. Aus diesem Grund wird eine Visualisierung in Form einer Webansicht selbst erstellt.

d3js

3 Konzeption

Dieses Kapitel befasst sich mit den theoretischen Überlegungen, welche in dieser Arbeit erarbeitet wurden. Konzepte bezüglich der Infrastruktur, Datenmodell, Entwicklungsumgebung und Visualisierung werden thematisiert. Es bildet somit die theoretische Basis, auf welcher der praktische Teil dieser Arbeit aufsetzt. Dieser wird in Kapitel 4 näher betrachtet.

3.1 Infrastrukturaufbau

Die folgende Grafik zeigt den Aufbau der Infrastruktur für diese Arbeit. Als Server kann hier beispielsweise ein Windows Server verwendet werden. Innerhalb dieses Servers befinden sich die Datenbank und der Webserver. Als Datenbank wurde hier Neo4j angedacht und als Webserver beispielsweise der Apache HTTP-Server. Der Webserver beinhaltet den entsprechenden Code mit den dazugehörigen Skripten.
Um auf diesen Inhalt zuzugreifen, muss zunächst der User an seinem Computer einen Browser öffnen. Dies kann wie hier abgebildet der Firefox sein. Anschließend wird über das Internet bzw. ein Intranet auf die Webseite zugegriffen.

Die einzelnen Komponenten sind in dieser Grafik nur beispielhaft aufgeführt. Diese können durch vergleichbare Komponenten ersetzt werden. Hier ein kleiner Auszug, welche gängigen Komponenten dies sein könnten:

- Server: Windows, Linux, Unix
- Datenbank: Neo4j, MySQL, PostgreSQL, CouchDB
- Webserver: Apache HTTP-Server, Tomcat, Nginx
- Internet / Intranet
- Browser: Firefox, Google Chrome, Internet Explorer, Apple Safari

Grafik

Abbildung 9: Infrastrukturkonzept

3.2 Datenmodellkonzept

Ein Datenmodell ist bei der Erstellung einer Datenbank sehr wichtig, da dies die einzelnen Verbindungen und Abhängigkeiten abbildet. Die folgende Grafik zeigt ein Beispiel für ein Datenmodell einer abstrahierten IT-Landschaft. Das Datenmodell wurde so erweitert, dass es möglichst realistisch ist und viele eventuellen Szenarien abdecken kann, welche die Entscheidungsfindung unterstützen.

Der Aufbau des Datenmodells ist in drei Spalten gegliedert. Die linke Spalte, welche in der Farbe Blau dargestellt ist, zeigt die Unternehmensstruktur. Bei der mittleren Spalte handelt es sich um das Informationssystem. Dieses ist die Verbindung zwischen der Unternehmensstruktur und der in der Farbe Grün dargestellten rechten Spalte, dem technischen Aspekt. In manchen Fällen kommt es vor, dass der Name der Relation sich wiederholt. Dies ist der Fall, wenn Verbindungen dieselbe Art der Beziehung aufweisen. Beispiele hierfür können der Abbildung 10 entnommen werden. Beziehungen zwischen Elementen innerhalb der Kategorie der Unternehmensstruktur (blau) sowie der Kategorie des technischen Aspekts (grün) sind zulässig. Des Weiteren werden auch Beziehungen innerhalb einer Objektklasse (Knoten) erlaubt. Als Beispiel sei hierfür eine Beziehung zwischen zwei Informationssystemen an dieser Stelle genannt. Diese könnte beispielsweise durch eine implementierte Schnittstelle hervorgerufen werden. Weiterhin sei erwähnt, dass eine Objektklasse sowie eine Relation mit Attributen erweitert werden kann. Dies hängt in der Realität von der entsprechenden Datengrundlage ab.

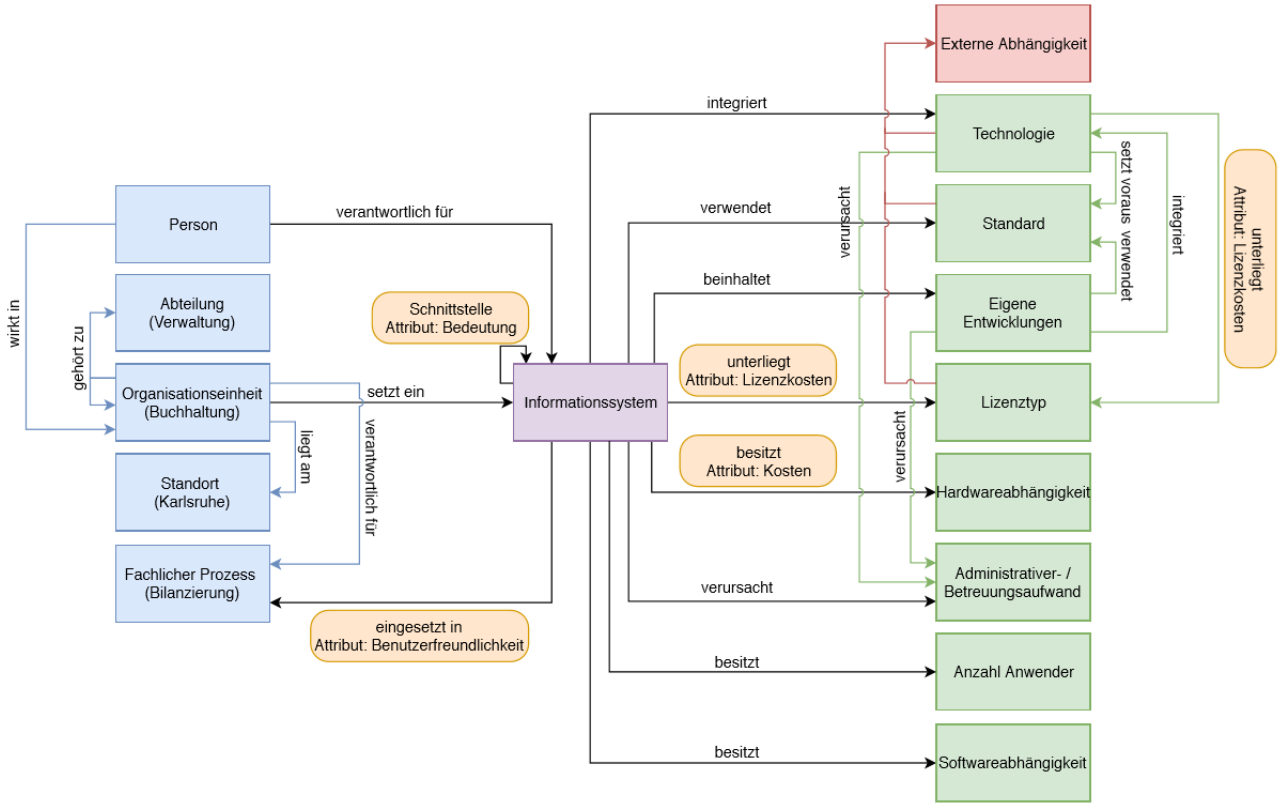

Abbildung 10: Datenmodell

Datenmodelle sind sehr aussagekräftig. Durch das visuelle Verständnis fungiert das Modell als eine Art Kommunikationstool zwischen Entwickler und Endanwender. Wenn man ein konzipiertes Datenmodell sieht, ist es zumeist sehr selbsterklärend und man kann die Zusammenhänge des Modells schnell erkennen, ohne thematisch tiefe Kenntnisse des Projektes zu haben.

Laut (Wirtschaftslexikon Gabler, 2018) beschreibt ein Datenmodell die zu verarbeitenden Daten eines Anwendungsbereichs mittels Grafik. Zudem zeigt dieses die Beziehung, welche die Daten untereinander aufweisen.

Durch ein Datenmodell können Redundanzen aufgedeckt werden. Ebenso wird dadurch ersichtlich, ob die Daten eindeutige Informationen aufzeigen. Unter Berücksichtigung dieser Punkte, können Rückschlüsse über die Datenqualität getroffen werden. Ein qualitativ hochwertiges Datenmodell ist somit ein essenzieller Bestandteil eines Softwareprojekts, welches Datenbanken als Ressource einbindet. Aus diesem Grund ist es wichtig sich vor der Implementierung mit dem Datenmodell auseinander zu setzen. Zusätzlich unterstützt der visuelle Entwurf der Datenstruktur, den Prozess der Implementierung.