

BACHELORTHESIS
Constanze Maaß

Konzeption und Umsetzung eines Knowledge Graphen zur Evaluierung des Einsatzes in der Nachweisdokumentation

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

Constanze Maaß

Konzeption und Umsetzung eines Knowledge Graphen zur Evaluierung des Einsatzes in der Nachweisdokumentation

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

Betreuender Prüfer: Prof. Dr. Stefan Sarstedt
Zweitgutachter: Prof. Dr. Kai von Luck

Eingereicht am: 18. Mai 2021

Constanze Maaß

Thema der Arbeit

Konzeption und Umsetzung eines Knowledge Graphen zur Evaluierung des Einsatzes in der Nachweisdokumentation

Stichworte

Wissensgraph, Wissensmanagement, Wissensrepräsentation, Inferenz, agil, Organisation, X-Team

Kurzzusammenfassung

Texte, Graphiken oder Bilder sind Informationsträger. Sie sind Gedankenstützen und Transportmittel, um Wissen für eine Gemeinschaft verfügbar zu machen. Die Abbildung 1.1 - siehe Kapitel Einleitung - zeigt eine graphische Aufbereitung der vorliegenden Theses. Durch die kategorisierten Informationen (Knoten) und die Pfeile (Kanten) bekommt die Abbildung einen Graphcharakter. Schnell sind die Abhängigkeiten der Konzepte rund um die Problemstellung "Dokumentenstapel" erkennbar und man kann eine erste Einschätzung darüber treffen, welche Wissensbestände zur Erfassung der Thematik vorliegen bzw. noch erarbeitet werden müssen. Es stellt sich also folgende forschungsleitende Frage: Was spricht dagegen, Informationen nicht nur als Texte abzulegen, sondern sie strukturell in einen Graphen einzubauen, um Wissen effektiv greifbar zu machen?

Constanze Maaß

Title of Thesis

Conception and implementation of a knowledge graph to evaluate the use in the verification documentation

Keywords

knowledge Graph, knowledge management, knowledge representation, inference, agile, organization

Abstract

Texts, graphics or images are information carriers. They are thought aids and Means of transport to make knowledge available to a community. The image 1.1 - see chapter “Einleitung” (Introduction) - shows a graphic preparation of the present thesis. The categorized information (nodes) and edges give the image the character of a graph. The dependencies of the concepts around the Problem of the “stack of documents” recognizable. What speaks against not just use text as place to keep information, but integrating it structurally in a graph, to make knowledge effectively available?

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellenverzeichnis	ix
Listings	x
1 Einleitung	1
1.1 Motivation	1
1.2 Problem	2
1.3 Ziel	3
1.4 Aufbau	3
2 Grundlagen	4
2.1 Knowledge/ Wissen	5
2.2 Knowledge Management (KM)	7
2.2.1 Ziele	7
2.2.2 Rahmenbedingungen	8
2.2.3 Aufgaben	10
2.2.4 Knowledge Management System (KMS)	12
2.2.5 Zusammenfassung	12
2.3 Nachweisdokumentation (NWD)	13
2.3.1 Team (T)	14
2.3.2 Systemakten (SA)	15
2.3.3 Zusammenfassung	16
2.4 Knowledge Graph (KG)	17
2.4.1 Architektur	17
2.4.2 Datenquellen (Data source)	18
2.4.3 Backend (Knowledge Extraktion und Fusion)	19
2.4.4 Datenbank (Knowledge Graph)	21

2.4.5	Frontend (Visualisierung)	23
2.4.6	Zusammenfassung	23
3	Analyse	24
3.1	Anforderungen	24
3.2	Sichten	26
3.2.1	Komponenten und Module	26
3.2.2	Module und Klassen	27
3.3	API-Spezifikation und Datenmodelle	27
3.4	Ontologie	28
4	Umsetzung	30
4.1	Ontologie	31
4.1.1	Implementierung	31
4.1.2	Herausforderung	33
5	Tests	35
6	Evaluierung	40
6.1	Auswertung der Umfrage	40
6.1.1	Einführung	40
6.1.2	Praxis	42
6.1.3	Bewertung	43
6.2	Zusammenfassung	46
7	Fazit	47
	Literaturverzeichnis	49
A	Anhang	51
A.1	API-Spezifikation und Datenmodelle	51
A.2	Tabellen	56
A.3	Ontologie SVG	58
A.4	Wordclouds	59
A.5	Umfrage	60
A.6	Inhalt der CD-ROM	66
	Selbstständigkeitserklärung	67

Abbildungsverzeichnis

1.1	Gedankenskizze zur Thesis	1
1.2	Gartners' Hype Cycle platziert den Höhepunkt des Interesses am Knowledge Graphen im Juli des Jahres 2020[7]	2
2.1	Ontologie der relevanten Konzepte und These	4
2.2	Original Grafik von Hugh MacLeod, überarbeitet von David Sommerville	5
2.3	Beispielhafte Erläuterung des Konzepts <i>Wissen</i> nach MacLeod	6
2.4	KM -Teilziele von innerbetrieblichen Abläufen über "Shareholder Value" (Anteilseigner eines Unternehmens) bis Erhöhung des Innovationspotenzials [4] (Quelle: eigene Darstellung)	8
2.5	Die Komplexität der Vernetzung von SAs mit verborgenen Verweisen auf anderen SAs (Quelle: Eigene Darstellung)	16
2.6	Semantischer Graph	17
2.7	Architektur eines Knowledge Graphen	18
2.8	Software-(Teil-)Lösungen	20
2.9	Graph und Adjazenzmatrix	21
2.10	Datenbankstrukturen	22
3.1	Anforderungen an das System	25
3.2	Komponenten und Module	26
3.3	Module und Klassen	27
3.4	Erwartete Ontologie mit zusätzlichen experimentellen Entitäten	28
3.5	Generierte Matrix mit default Ontologie	29
4.1	Implementierte Benutzer-Oberfläche zur Visualisierung des Knowledge Graphen	30
4.2	Generische Lösung für die Benennung der Relationen (Prädikate)	31
5.1	Test - Handeln unterschiedlicher Dateiformate und Extraktionsschritt	35

5.2	Test - Verbindung von Team Kranken und Team Dragonfish über ihre Software-Ressourcen	36
5.3	Anforderung- Graph filtern mit bestimmten Begriffen	37
5.4	Anforderung- Graph-Darstellung ändern	38
6.1	Bewertung der Funktionalität des KG-Systems	42
6.2	Der Einsatz eines Knowledge Graphen im Wissensmanagement könnte ...	44
6.3	Mit dem Einsatz eines Knowledge Graphen im Wissensmanagement ist es möglich	45
6.4	Wie bewertest du den Einsatz eines Knowledge Graph Systems in der Dokumentation (Projekt- / System- / ect) ,	46
7.1	Hume Technologie	48
A.1	Wordclouds Beispieldaten	59

Tabellenverzeichnis

2.1	Disziplinen von Knowledge Management nach Denham Grey (Quelle: eigene Darstellung)	7
2.2	Strukturen von Datenquellen	19
6.1	Anmerkungen zur Motivation	41
6.2	Anmerkungen zur Einleitung	42
A.1	Ergebnis Umfrage (Bewertung-Aussagen I)	56
A.2	Ergebnis Umfrage (Bewertung-Aussagen II)	57

Listings

4.1	Ontologie Implementierung	31
4.2	Default Prädikate der These	32
A.1	Rest Api Dokumentation	51
A.2	SVG Ontologie Code - Auszug	58
A.3	README Markdown Datei mit dem Inhalt der CD-ROM	66

1 Einleitung

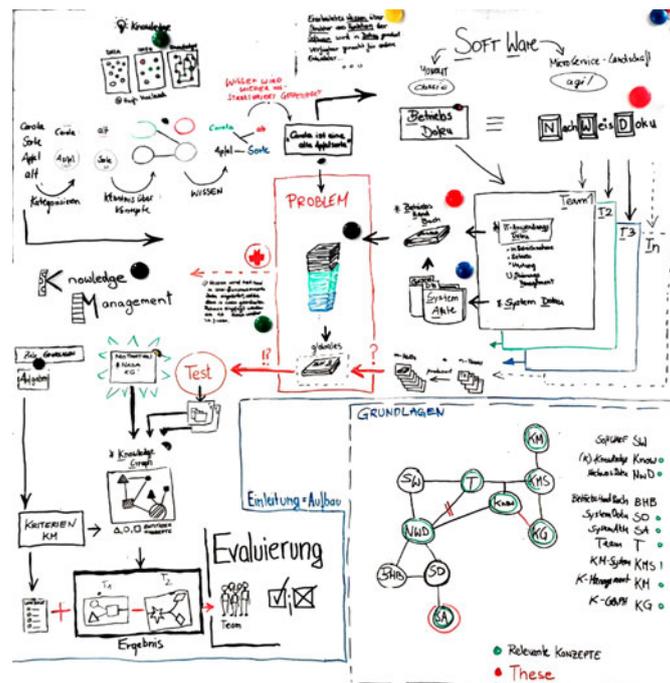


Abbildung 1.1: Gedankenskizze zur These

1.1 Motivation

Aktuell lässt sich die Tendenz ausmachen, dass die Branche für maschinelles Lernen den "Knowledge Graph" (KG) als Schwerpunktthema in der KI einstuft. Gartners "Hype Cycle" ordnet sie im Juli 2020 auf den Höhepunkt des Interesses ein - siehe Abbildung 1.2.

Beispielsweise plädierte der "Chief Knowledge Architect" der NASA, David Meza, 2017 für den Einsatz einer "Lessons Learned" Datenbank, welche angelehnt an die KG-Technologie

die Masse an produzierten Daten für das menschliche Gehirn zusammenhängend darzustellen vermag. Meza erklärte, seine Hauptmotivation als Projektmanager liege darin, das gesamte System zu verstehen. Gesamtsystem bedeutet das Zusammenspiel aller Komponenten. Beispielsweise arbeiten mehrere Teammitglieder an unterschiedlichen Teilsystemen wie Batterie-, Kraftstofftank- oder Blechtungs-Subsystem. Alle leben in ihrer eigenen Welt und sprechen normalerweise nicht miteinander. Meza stellt fest, ein Manager müsse dazu in der Lage sein, diese Informationen durchzusehen und diese zusammenzuführen.

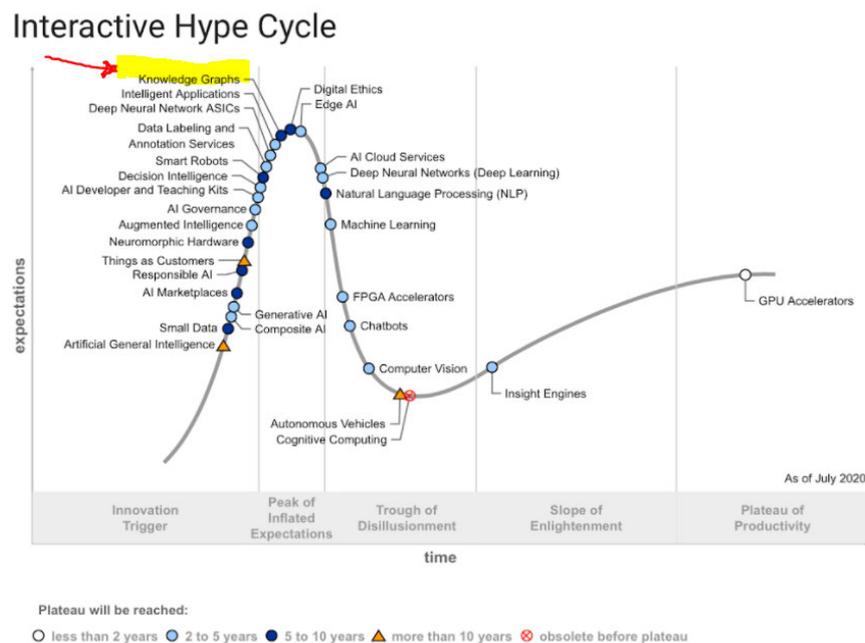


Abbildung 1.2: Gartners' Hype Cycle platziert den Höhepunkt des Interesses am Knowledge Graphen im Juli des Jahres 2020[7]

1.2 Problem

Verteilte Systeme finden sich auch in der agilen Softwareentwicklung wieder. Auch hier sollten Business Designer, Business Architekten oder Technische Architekten einen Gesamtüberblick über die Systeme haben, um qualitativ gute Entscheidungen zu treffen. Von der Priorisierung von Aufgaben bis hin zum Refactoring der Architektur ist eine Übersicht von Abhängigkeiten fundamental wichtig. Doch stellt sich in der Praxis nur

zu oft heraus, dass dieser wichtige Überblick mehr als ein “nice to have” ist. Es sollte als ein notwendiges Ziel begriffen werden. Das Wissen endet an den System- bzw. Domänengrenzen, doch Entscheidungen werden auf einer höheren Ebene getroffen. Damit ist die Qualität der Entscheidungen nicht unbedingt gewährleistet

1.3 Ziel

Das Ziel dieser Arbeit ist es, in einem agilen Softwareprojekt auf unterster Ebene, Wissen bereit zu stellen. Hierfür sollte mit dem Einsatz eines Knowledge Graphen (KG) möglich sein, Teile der Nachweisdokumentation automatisiert miteinander zu verknüpfen. Eine Visualisierung der Beziehungen zwischen Informationen nähert sich dann der Definition von Wissen an. Die Konzeption des KGs strebt dabei die Einhaltung von Kriterien des Knowledge Management (KM) an, um ein geeignetes System für die Dokumentation zu etablieren. Der Inhalt wird von der Nachweisdokumentation bestimmt und fokussiert sich auf eine Darstellung der Systemkomponenten von zwei Teams. Rahmenbedingungen und Ziele des Systems werden hier auch am KM ausgerichtet, um abschließend den Einsatz bewerten zu können.

1.4 Aufbau

Im ersten Kapitel werden die relevanten Konzepte dieser Arbeit näher betrachtet und zueinander in Beziehung gesetzt. Vom eigentlichen Wissen über die Kriterien und Aufgaben von Wissensmanagement spannt sich ein Bogen bis hin zu der Frage der Wissensauswahl: Welches Wissen aus der Nachweisdokumentation soll im Knowledge Graphen (KG) überhaupt dargestellt werden? Aus diesem Grundlagenkapitel lassen sich dann Anforderungen für den KG in der Analyse herausarbeiten. In der Umsetzung sollen relevante Implementierungen näher erläutert werden. Nach den Tests gegen die bestehende Software mit ihren Funktionen, sollte diese dann auch von zwei Test-Teams geprüft und bewertet werden. Abschließend sollte eine Evaluierung des Einsatzes möglich sein.

2 Grundlagen

Um einen einheitlichen Überblick des umfassenden Themengebiets “Wissensmanagement” zu erhalten, werden in den folgenden Abschnitten die Oberbegriffe Wissen, Nachweisdokumentation, Knowledge-Management und -Graph mit Arbeitsdefinitionen umrissen und spezifiziert. Die Abbildung 2.1 zeigt eine Ontologie der relevanten Konzepte. Diese sind grundlegend für die Evaluierung des Einsatzes einer Knowledge Graph Technologie. Aus den gewonnenen Informationen sollten sich dann Anforderungen für den KG in der Analyse formulieren lassen.

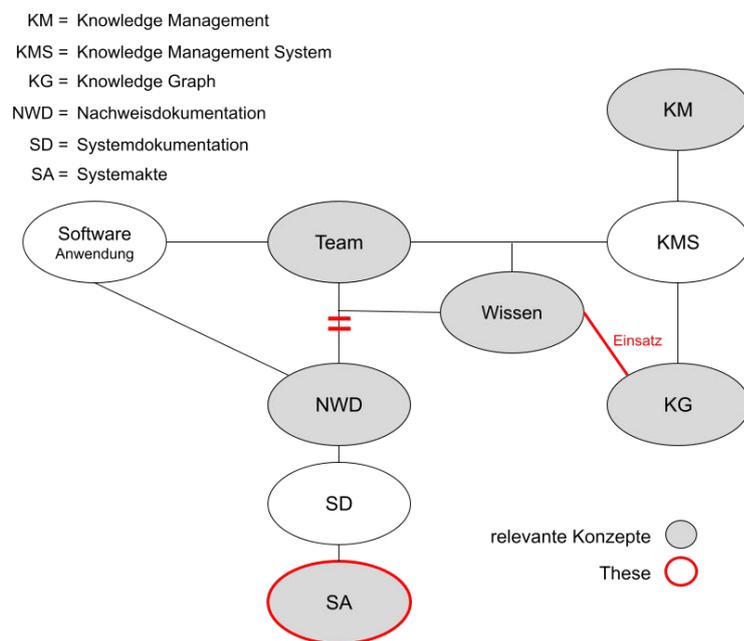


Abbildung 2.1: Ontologie der relevanten Konzepte und These

2.1 Knowledge/ Wissen

Es existiert eine Vielzahl an Wissen-Definitions-Ansätzen. Eine Arbeitsdefinition, angelehnt an die schematische Darstellung von Hugh MacLeod (Abbildung 2.2), soll die Komplexität des breiten philosophischen/wissenschaftlichen Themengebiets des Wissens eingrenzen. Des Weiteren greift die Definition Aspekte der Semantik auf.

Arbeitsdefinition: Wissen ist die Kenntnis über die Beziehungen zwischen Konzepten und Objekten der realen Welt und ihrer Bedeutungen.

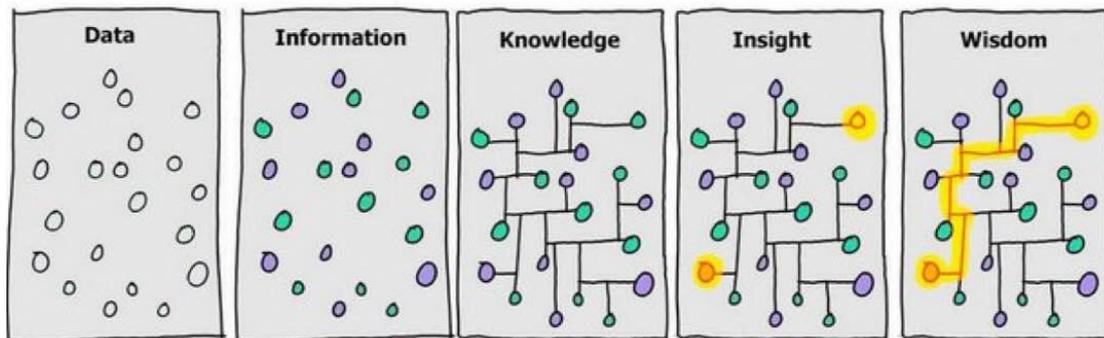


Abbildung 2.2: Original Grafik von Hugh MacLeod, überarbeitet von David Sommerville

Das Konzept Wissen aus der Abbildung 2.2 kann auf folgendes Beispiel übertragen werden (Abbildung 2.3). Im Fokus der Betrachtung stehen “Apfel”, “Carola” und “Sorte”. Diese Begriffe bleiben für Menschen ohne Kenntnis der Konzepte hinter den Worten einfache Wörter. Menschen, die diese Worte Konzepten zuordnen können, ist ein “Apfel” ein Obst, eine “Sorte” eine Art und “Carola” ein Name. Als wissender Mensch wird also jemand bezeichnet, der diese Informationen in Beziehung setzen kann und sich erschließt, dass eine Apfelsorte namens “Carola” existiert.

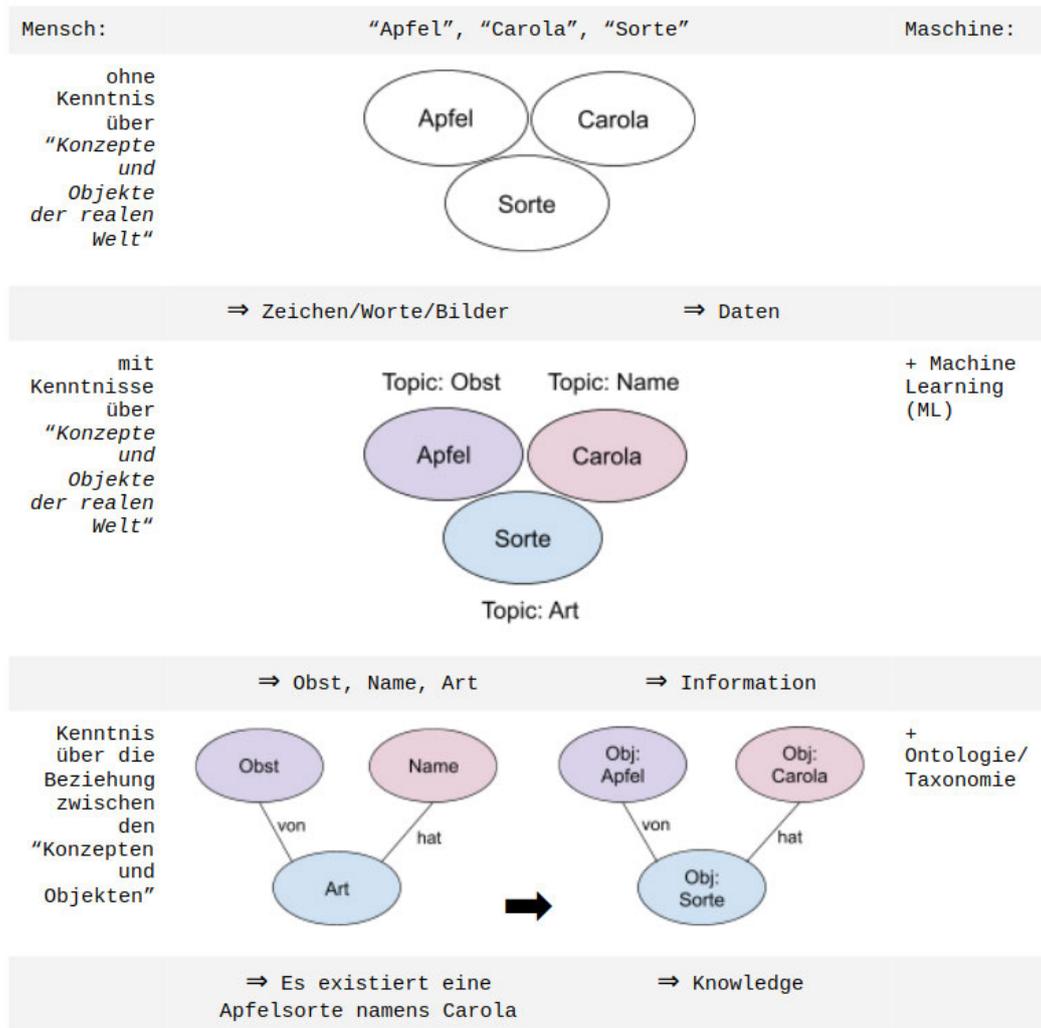


Abbildung 2.3: Beispielhafte Erläuterung des Konzepts *Wissen* nach MacLeod

2.2 Knowledge Management (KM)

In der Wissenschaft existiert keine eindeutige Definition von Knowledge Management (Wissensmanagement). Es werden Eigenschaften und Methoden genannt, über die sich die Experten einig sind. So spezifizierte Denham Grey KM wie folgt:

Broad-based discipline: mine (analyze) data; capture, create, store, catalog, validate, transform, and disseminate/share information; capture, emerge, store, catalog, and disseminate/share knowledge. [4]

Kurz: KM ist eine Disziplin, die sich mit Methoden der Verarbeitung, Speicherung und Bereitstellung von Daten, Informationen und Wissen beschäftigt.

Die in der Tabelle 2.1 aufgeführten Methoden/Aufgaben von Knowledge Management bilden die Basis für diese Arbeit und werden im Abschnitt 2.2.3 differenziert betrachtet. Knowledge Management ist eine Disziplin, die sich mit einer Reihe von Aufgaben, hauptsächlich der Daten-, Informations- und Wissensverarbeitung, sowie -bereitstellung und -speicherung beschäftigt. Dabei folgen diese Aufgaben einem übergeordneten Ziel und fundamentalen Rahmenbedingungen, welche in den Abschnitten 2.2.2 und 2.2.1 aufgeführt werden.

Wissens Management		
Daten	Informationen	Wissen
analysieren	katalogisieren	sammeln
erfassen	validieren	hervorheben
erstellen	transformieren	hervorheben
speichern	teilen	katalogisieren
		teilen

Tabelle 2.1: Disziplinen von Knowledge Management nach Denham Grey (Quelle: eigene Darstellung)

2.2.1 Ziele

Da KM größtenteils in Organisationen stattfindet, ist das globale Ziel von KM ein nachhaltiger Wettbewerbsvorteil für das jeweilige Unternehmen/Organisation. Um diesen

Vorteil zu erreichen, sollten unterschiedliche Teilziele anvisiert werden, siehe Abbildung 2.4.¹.

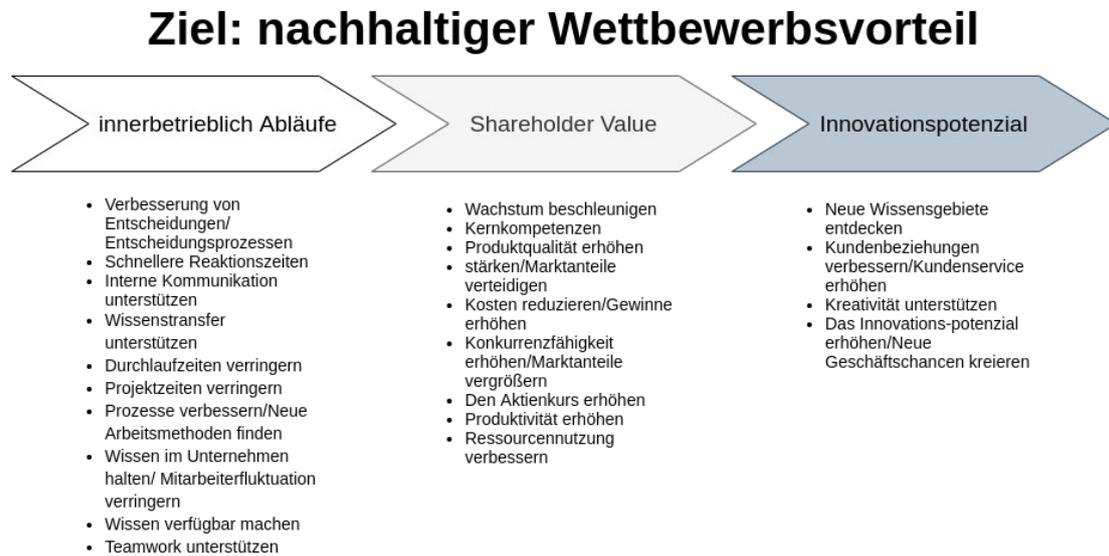


Abbildung 2.4: KM -Teilziele von innerbetrieblichen Abläufen über “Shareholder Value” (Anteilseigner eines Unternehmens) bis Erhöhung des Innovationspotenzials [4] (Quelle: eigene Darstellung)

Das Vorgehen von KM ist bestimmt durch das Zusammenspiel aus Erwartungen an Produkt, Produktivität, Innovation, Teamwork und Kosten. Diese Erwartungen können als Maß für den Erfolg von KM betrachtet werden.

Der Einsatz eines Knowledge Graphen in der Nachweisdokumentation soll in der Evaluierung mit diesen Erwartungen betrachtet werden.

2.2.2 Rahmenbedingungen

Maßgeblich entscheidend für den Erfolg von KM ist die innere Einstellung organisationsinterner Wissensträger zu den angestrebten Zielen der KM. Wissensträger sind alle Managementebenen bis zum einzelnen Mitglied einer Organisation. Hüttenegger fasst

¹[4]. Vgl. Seite 10.

sieben Punkte für eine optimale Grundlage von KM zusammen, die im Folgenden aufgezählt und inhaltlich umrissen werden.²

Der Wille zu KM und das Wissen über das Warum - Die Wichtigkeit und die Vorteile von KM sollten bekannt sein und gelebt werden. Eine gemeinsame Vision soll zu zielorientierten Problemlösungen führen und das Zusammenarbeiten begünstigen.

Eine Kultur des Teilens und der Zusammenarbeit - Gegenläufig zur Kultur des Teilens ist die Motivation, dass das Wissen beim Wissensträger eingekapselt wird - zum Beispiel aus Angst um den eigenen Arbeitsplatz. Deshalb setzt das KM ein hohes Maß an Vertrauen voraus, das auf der organisatorischen Ebene berücksichtigt werden muss. Auch ein KMS kann nach Hütteneggers Einschätzung keine Abhilfe in diesem Problemkontext schaffen.

Intrinsische versus Extrinsische Motivation - Unter diesem Gegensatzpaar versteht man den schmalen Grad zwischen Motivation, Opportunismus und Demotivation. Hier entscheidet die innere (intrinsische) Motivation unter Einflussnahme der äußeren (extrinsischen) Motivation den Erfolg von KM. Dabei kann die extrinsische Motivation negativ oder positiv konnotiert sein, wobei letztere Prägung langfristig wirkt. Hüttenegger gibt hierfür beispielhaft Bestrafung und Anerkennung an.

Wahrhaftigkeit - Dabei geht es um das generelle Problem, korrekte und zeitgerechte Informationen zu erhalten. Die Nutzer müssen Vertrauen in die Richtigkeit der Inhalte haben. Eine Lösung bietet beispielsweise das automatische Einfügen und Aufbereiten von Daten durch generierte System-Berichte.

Ziele für KM setzen und überprüfen - Es wird empfohlen, die gesetzten Ziele regelmäßig, beispielsweise durch Umfragen, Überprüfung der Durchlaufzeiten oder der Projektlaufzeiten zu monitoren, um herauszufinden, ob die Kosten der KM -Initiative gerechtfertigt sind. Als aussagekräftige Indikatoren für den Erfolg von KM könnten dementsprechend kürzere Laufzeiten und eine hohe Anzahl an Umfrage-Teilnehmenden gelten. Die Ziele sind dabei individuell an die Gegebenheiten der Organisation anzupassen und meist langfristiger Natur.

KM und Innovation/Kreativität - Ein effektives KM-System setzt den Fokus auf vorhandene Vorgehensweisen und gewohnte Technologien und kann damit Innovationen behindern. Deshalb schlägt Hüttenegger die Kategorisierung von Informationen innerhalb der KM vor, damit Zeit und Aufmerksamkeit auf relevante Inhalte gelenkt werden.

²[4] Vgl. Seite 13ff.

Barrieren und Hindernisse: Entfernen und Errichten - Um den Informationsfluss effektiv gestalten zu können, müssen Barrieren und Hindernisse effektiv beseitigt werden, ohne dabei ihre Schutzfunktion zu opfern. Die richtige Herangehensweise ist auch hier eine Einzelbetrachtung der Elemente.

Zeit, Geld, Strukturen und Traditionen - Hüttenegger geht hier auf besonders wichtige organisatorische Aspekte in Zusammenhang mit der Entwicklung und Nutzung einer KM ein. Er gibt zu bedenken, dass auch nach der Einführung des KM genug Budget und Personal für die optimale Nutzung und Weiterentwicklung des Systems vorhanden sein sollte. Außerdem müssen effektive Strukturen der Zusammenarbeit in der Organisation geschaffen werden, beispielsweise durch (auch virtuelle) Gemeinschaften. Damit die KM angenommen wird, sollten Tradition und aktuelle Struktur als Basis für Veränderungen berücksichtigt werden, um alle Mitarbeiter einzubeziehen.

Fundamental ist dementsprechend eine organisationsweite Transparenz der angestrebten Teilziele. Mitarbeiter könnten dann Probleme mit Blick auf eine gemeinsame Vision von KM lösen.

2.2.3 Aufgaben

Nur ein integriertes Gesamtsystem kann laut Hüttenegger die Anforderungen und Ziele von KM abdecken. Die fachlichen Anforderungen werden im Folgenden zusammengefasst:

Erfassen- Das Erfassen und Einordnen von Informationen ist eine zentrale Aufgabe der KM. Dies sollte, so weit wie möglich, automatisiert erfolgen und Aspekte wie Kontext, Historie und Qualifikation der Informationen beinhalten.

(Technisch) Integrieren und Einbinden (Verbinden) - Ein vollständiges KM -System muss Zugriff auf alle Informationen und Daten haben. Doppelstrukturen mit nicht-integrierten Systemen können durch bidirektionale Verbindungen verhindert werden. Das Ziel dieser Integration und Verbindung ist die Verringerung der Durchlaufzeiten.

Transformationen - Transformationen sind einerseits für Daten aus externen Quellen und andererseits für bidirektionale Verbindungen notwendig. Die KM muss auf Daten und Informationen effektiv zugreifen oder diese müssen so verändert werden, dass sie in den eigentlichen Datenspeicher der KM eingefügt werden können.

Speicherung - Für ein vollständiges KM System ist die Integration vorhandener Datenbanksysteme wie DBMS und DMS essentiell. Es sollte Funktionalitäten für alle verwalteten Elemente anbieten und unterstützen. Hüttenegger benennt Anmerkungen, Funktionskontrolle und Änderungsprotokolle als fundamentale Funktionen. Er verweist zudem darauf, dass Redundanzen und Duplikationen grundsätzlich zu vermeiden bzw. zu reduzieren sind.

Flexibles und adaptierbares Berechtigungssystem - Die Vertraulichkeit der Daten und die Sicherstellung gesetzlicher Anforderungen sind ein Eckstein für ein flexibles und anpassbares Berechtigungssystem. Gleichzeitig muss legitimen Nutzern (auch von außerhalb der Organisation) Zugriff auf alle relevanten Informationen gewährt werden. Als optimale Grundlage dafür dient eine zentrale Benutzerdatenbank wie ein "Directory Service" in Verbindung mit anderen Elementen außerhalb der KM-Struktur.

Leute miteinander verbinden - Das KMS sollte in der Lage sein, zwischenmenschliche Kommunikation zu protokollieren und zu strukturieren. Beispielsweise kann das KM Gesprächsmitschnitte, E-mails, Diskussionsforen oder Videokonferenzen durch "Speech Mining" zugänglich machen, um Teams, Organisationen, Gruppen, etc. in Kommunikationsprozessen zu unterstützen.

Erfassen und Verwaltung und Präsentation von Kontext, Links und Historie - KM - Systeme müssen Kontext, Links und Historien nach individuellem Bedarf und nach dem Maßstab höchster Flexibilität im Zugriff gestalten. Zudem muss das KM Mehrdeutigkeit auf allen Ebenen unterstützen.

Automation - Notwendige Aufgaben werden routinemäßig durchgeführt oder angestoßen. Beispielhaft dafür stehen: : Nachrichten klassifizieren, Workflow-Prozesse optimieren.

Suchen, Finden und Zugreifen - Diese Funktionen sind aus Nutzerperspektive die Sichtbarsten und Wichtigsten des KMS. Dafür ist es notwendig, dass das System Zugriff auf alle gespeicherten externen sowie internen Informationen hat in nutzergerechter Form gleichzeitig aufbereitet und darstellt. Diese Anforderungen summiert Hüttenegger unter den Begriffen der Transparenz und Nachvollziehbarkeit. Dennoch bezeichnet er erstere als einer der Schlüsselfaktoren eines KM, deshalb mit organisatorischen wie technologischen Maßnahmen adressiert werden sollte. Unter Nachvollziehbarkeit versteht man ein komplexes Konzept, das eng mit der Historie und Transparenz sowie anderen Elementen verbunden ist. Benutzer des Systems sollen zu jedem Zeitpunkt verstehen können, was mit einzelnen Informationen geschieht.

Unverzichtbare Basis - Nur ein System, das schnell und zuverlässig funktioniert und eine einfache Bedienung anbietet, wird genutzt. Diese Benutzerfreundlichkeit erfordert meist Kompromisse beim User Interface, das sowohl auf erfahrene als auch unregelmäßige Nutzer*innen zugeschnitten sein sollte. Sind diese beiden Aspekte nicht gegeben, wird das KMS scheitern.

2.2.4 Knowledge Management System (KMS)

Ein Knowledge Management System in der IT ist eine Software/Anwendung, die die Aufgaben von KM für sich als Anforderungen definiert. [4]

Softwarelösungen sind derzeit Dokumentenmanagement-Systeme (z.B. Atlassian Confluence), Groupware-Anwendungen (z.B. Microsoft SharePoint) oder Content Management Systeme. Ihr Ansatz einer hierarchischen Dokumentenablage ist ein Abbild der realen Welt. Ihre Strukturen erinnern an Verzeichnisse und Aktenschränke. Erarbeitetes Wissen wird auf unterschiedliche Weise hinterlegt, in Tabellen oder Datenbanken strukturiert, in Dateiformaten wie XML, JSON, HTML etc. semi-strukturiert oder unstrukturiert in Texten oder Bildern abgelegt.

Vergleicht man diese Daten mit der Wissensdefinition, haben sie maximal den Status kategorisierter Informationen inne. Denn filtert man in der Klartext-Suche den Inhalt nach einem Konzept oder einer Information, wird im besten Fall nur dieses Gesuchte ausgegeben. Das Wissen über umliegende, ebenso relevante Konzepte und Informationen bleibt verborgen. Diese Verknüpfung von Informationen ist eine Anforderung, die durch den Knowledge Graphen verwirklicht wird.

2.2.5 Zusammenfassung

Zusammenfassend kann man eine Reihe von Faktoren und Anforderungen nennen, die das Gelingen einer KM -Initiative beeinflusst. Diese Faktoren sollen bei der Implementierung des Knowledge Graphen berücksichtigt werden. Ebenso stellen die Rahmenbedingung für KM optimalerweise einen Filter für die Evaluierung des KG-Einsatzes dar. Beispielsweise ist eine Einführung in das System für eine geteilte Vision von KM sicherlich vorteilhaft.

2.3 Nachweisdokumentation (NWD)

Der Begriff Nachweisdokumentation ist in der IT weniger geläufig. Er wird kurz im Praxisbuch - IT Dokumentation erwähnt. NWD wird darin als Dokumentation für unterschiedliche Stakeholder wie IT-Administrator oder IT-Support umschrieben, also der IT-Betrieb³. In dieser Arbeit nimmt ein Team die Rolle des Adressaten und Absenders ein und wird in Abschnitt 2.3.1 spezifiziert. Da in der agilen Softwareentwicklung wenig bis keine Richtlinien, Bestimmungen oder Definitionen im Hinblick auf Dokumentation geliefert werden - da ihr keine hohe Priorität zugemessen wird [siehe Agiles Manifest] - orientiert sich diese Arbeit an dem Aufbau eines Betriebshandbuchs (BHB) der Betriebsdokumentation (BD).⁴ Es wird eine Differenzierung zwischen BD und NWD vorgenommen, da in der agilen Softwareentwicklung die einzelnen fachlichen Domänen mit ihren Microservices (MS), und nicht das gesamte System bzw. die gesamte Anwendung dokumentiert werden.

Der Inhalt eines BHB wird - nach Reiss⁵ - wie folgt gegliedert:

1. Beschreibung der Anwendung
- 2. Technische Beschreibung *Systemdokumentation**
3. Installation und Konfiguration
4. Verfügbarkeit
5. System- und Anwendungsbetrieb
6. Sicherheit
7. Störungsmanagement

Die "Technische Beschreibung" ist hier eine Form von Systemdokumentation (SD). Sie umfasst eine Reihe von Erläuterungen, etwa die der Systemkomponenten, Plattform, Anforderung an den Produktivbetrieb und des Testbetriebs, der externen und internen Schnittstellen und ihrer Datenflüsse.⁶ Diese Beschreibungen können als Systemakten (SA) in der SD angelegt werden.

³[6] Vgl Seite 2

⁴[6] Vgl Seite 185.

⁵[6] Vgl Seite 185.

⁶[6] Vgl Seite 183.

Systemakten bezeichnen den Aspekt der NWD, der im angestrebten Knowledge Graphen abgebildet werden soll. Diese werden im Abschnitt 2.3.2 näher erläutert. Alle anderen Inhalte des BHB werden im Rahmen dieser Arbeit nicht näher betrachtet.

2.3.1 Team (T)

Das Team ist in der agilen Softwareentwicklung ein sogenanntes Cross-Funktionales-Team (X-Teams). Es bildet einen vertikalen Querschnitt der Organisationsstruktur ab und befasst sich mit der Realisierung einer Teilanforderung der gesamten Anwendung [8].

Dieses Team ist verantwortlich für die NWD ihrer Domäne. Die Umsetzung findet in einem selbst definierten Rahmen statt. So kann sich die SD auf unterschiedliche Softwaretools verteilen:

- SA werden beim Web-Service-Provider (z.B. AWS) bereitgestellt.
- API-Dokumentation wird als Swagger-Datei direkt im Microservice integriert.
- Anforderungen an den Produktions-Betrieb und Test-Betrieben werden in Text-Dokumenten festgehalten (z.B. Confluence ein Dokumenten-Management-System DMS).
- die Links zu den einzelnen Bestandteilen der Systemdokumentation werden im DMS hinterlegt.

Die Verteilung des Wissens nimmt mit der Anzahl der Teams zu, so dass ein Gesamtbild der Systemlandschaft auf manuelle Weise nicht mehr wartbar ist. Hier sollte ein Automatismus eingebaut werden, der das Zusammenführen der Informationen koordiniert. Denn ein Regelwerk an Dokumentationsvorschriften lässt sich in der agilen Gemeinschaft nicht etablieren da es im Agilen Manifest heißt: “funktionierende Software vor Dokumentation”.⁷

Eine solche Zusammenführung und Darstellung von Informationen könnte zu besseren Entscheidungen führen. Und genau darin besteht ein Teilziel von KM (siehe Abschnitt 2.2.1).

⁷[1] [Agiles Manifest](#).

2.3.2 Systemakten (SA)

Systemakten sind eine Form von modularem, hierarchischem Aufbau der Systemdokumentation. Sie können in Gruppen unterteilt werden: IT-Systeme (SYS), Anwendung (APP), Netzwerk und Kommunikation (NET), Industrielle IT (IND) und Infrastruktur (INF).⁸ Der BSI (Bund für Sicherheit in der Informationstechnik) unterteilt diese Haupt - Baustein - Gruppen in 100 weitere Bausteine⁹. Die komplexe Informationsmenge lässt sich in abgeschwächter Form auf die der agilen Softwareentwicklung übertragen.

Teams, die eine Teilanforderung der gesamten Anwendung umsetzen, sollten die Inhalte für ihre Domäne liefern, um eine vollständige Systemdokumentation zu gewährleisten. Sinnvolle Inhalte für Systemakten der fünf Schichten (SYS, APP, NET, IND und INF), werden durch Reiss beschrieben.¹⁰ Im Rahmen dieser Arbeit werden die SYS- und APP-Inhalte näher betrachtet.

Die Inhalte dieser beiden Gruppen zeigen eine direkte Abhängigkeit voneinander: durch den Verweis vom APP-Punkt "Technische Spezifikation der Systemkomponenten" auf entsprechende SYS - Systemakten. In der agilen Softwareentwicklung sind die SYS-Inhalte der Teams stark miteinander vernetzt. Durch die Verweise auf Systemkomponenten/ Kommunikationskanäle anderer Teams, ist hier ein hohes Maß an Rechercheaufwand nötig, um ein Gesamtbild der Systeme zu erhalten. Diese vernetzte Struktur, am Ende einer hierarchischen Struktur, ist in Abbildung 2.5 dargestellt.

Relevant für ein Team ist ein globales und einheitliches Verständnis für die Systemlandschaft, um die Systeme weiter zu entwickeln und sie wartbar zu halten. So sollten bei der vorangegangenen Betrachtung der Systemdokumentation die Verweise zwischen Systemakten klar sichtbar sein. Dies wird meist durch manuell erstellte Schaubilder realisiert, welche in Dokumenten hinterlegt werden. Das Team definiert das Update dieses Schaubildes als Regel, die allerdings in den meisten Fällen verletzt wird. Dieses Versäumnis kann unterschiedlichen Faktoren geschuldet sein, z.B. Unkenntnis der Existenz bzw. des Dokumentenablage- Orts oder des Vergessens aufgrund zeitlicher Engpässe. Dieses Unterlassen des Updates verletzt den an die Dokumentation gerichteten Anspruch der "Wahrhaftigkeit" eine der Rahmenbedingungen aus Abschnitt 2.2.2 für KM. Dieser Aspekt der korrekten und zeitgerechten Lieferung der Information soll sich als Anforderung an den Knowledge Graphen wieder finden.

⁸[6] Vgl Seite 100.

⁹[2] [BSI Bausteine 2021](#).

¹⁰[6] Seite 120 ff Tabelle 4.1.

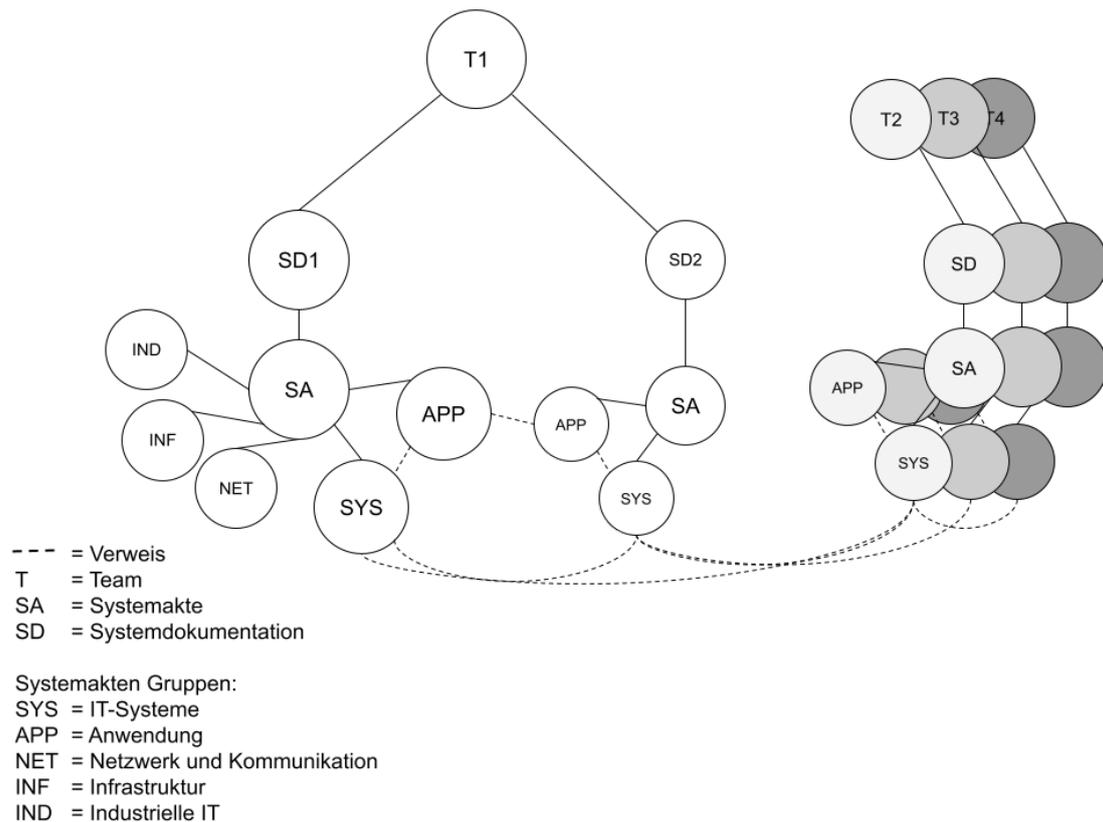


Abbildung 2.5: Die Komplexität der Vernetzung von SAs mit verborgenen Verweisen auf anderen SAs (Quelle: Eigene Darstellung)

2.3.3 Zusammenfassung

Die Nachweisdokumentation beinhaltet Systemakten. Diese sollen die Teilsysteme von sogenannten Cross- Teams beschreiben. NWD folgt zur Wiederfindung von Informationen einer hierarchischen Baumstruktur mit wichtigen Einfüge-Regeln. Diese Archivierungsform wird zurecht Dokumentation genannt, denn es handelt sich dabei um das Verwalten von Dokumenten und nicht von Wissen. Die eigentlichen Beziehungen zwischen den Informationen (Systemen) sind in den Dokumenten als Vermerk oder Links hinterlegt. Durch aktives Nachgehen der Verweise lässt sich eine Gesamtbild des Systems zeichnen. Laut Definition besteht aber Wissen in der der Kenntnis über die Beziehungen von Objekten. Folglich sollte der Einsatz des Knowledge Graphen (KG) in der NWD die Beziehungen zwischen den Systemen offen legen/ visualisieren und eine Anforderung an den zu implementierenden KG darstellen.

2.4 Knowledge Graph (KG)

Ein Knowledge Graph, auch semantischer Graph genannt (siehe Abbildung 2.6), ist eine Wissensbasis, die ein strukturiertes Datenmodell (Ontologie/ Taxonomie) oder eine Topologie zur Integration von Daten verwendet. Wissens - Graphen werden verwendet, um miteinander verknüpfte Beschreibungen von Entitäten - Objekten, Ereignissen, Situationen oder abstrakten Konzepten - in einer Form Semantik- wie Subjekt, Prädikat, Objekt - zu speichern.

Google Search bedient sich dieser Technologie, um seinen Nutzern relevante Nebeninformationen ihrer Suche anzeigen zu lassen. Diese Funktionalität ist für die NWD erstrebenswert, da beispielsweise die Nachbar- Systeme ebenso relevant für die Fehlerrecherche sind.

Die folgenden Abschnitte befassen sich mit der Architektur und der Funktionsweise eines Knowledge Graph Systems. Dabei wird Bezug auf die herausgearbeiteten Anforderungen an einen KG genommen.

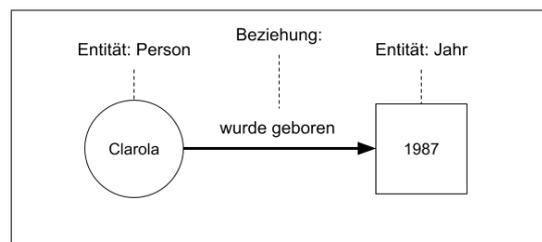


Abbildung 2.6: Semantischer Graph

2.4.1 Architektur

Der Prozess, Wissen aus Datenquellen zu extrahieren, spiegelt sich in der Architektur des KGs wieder: Vom Extrahieren der Entitäten über die Fusion von Relationen und Entitäten in eine Ontologie, bis hin zur Datenspeicherung- und darstellung in Graphenform. Der Aufbau eines KGs wird im Artikel "Architecture of Knowledge Graph Construction Techniques" beschrieben [9]. Die Abbildung 2.7 zeigt fünf Komponenten und die Datenquelle. Die Komponenten Knowledge - Extraktion, -Fusion, -Graph, -Retrieval, Visual

Display und deren Funktionen können den Oberbegriffen Frontend, Backend und Datenbank zugeordnet werden. Eine solche Aufteilung und Funktionsweise lässt sich auf den Entwurf des KGs übertragen.

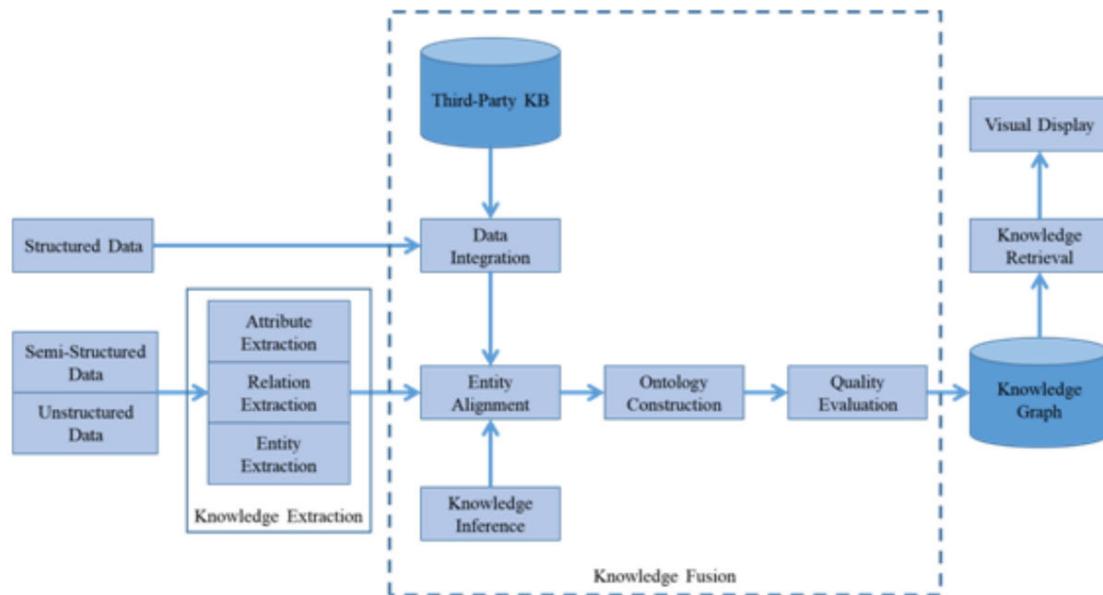


Abbildung 2.7: Architektur eines Knowledge Graphen

Im Folgenden werden die Aufgaben und Funktionsweisen der Komponenten des Knowledge Graphen näher erläutert.

2.4.2 Datenquellen (Data source)

Daten aus unterschiedlichen Datenquellen liegen in strukturierter, semi strukturierter und unstrukturierter Form vor. Beispiele für die Struktur-Formaten lassen sich aus der Tabelle 2.2 entnehmen.

Eine Aufgabe des KGs besteht darin, die Informationen aus den diversen Strukturen herauszuarbeiten und diese dann in eine effektiv lesbare Form - für Mensch und Maschine - zu übertragen. Dies soll in der Komponente Backend realisiert werden.

		Datenquellen		
		strukturierte Daten	semistrukturierte Daten	unstrukturierte Daten
		.csv	.json	.txt
		.db	.xml	.doc
Bsp		.xls	.html	.jpg
Formate		.dbs	.svg	.png
			.css	.pdf

Tabelle 2.2: Strukturen von Datenquellen

2.4.3 Backend (Knowledge Extraktion und Fusion)

Die Aufgaben des Backends bestehen darin, die unterschiedlichen Daten zu lesen, verarbeiten, speichern und abrufbar zu machen.

Der Aufgabenbereich *Lesen* schließt die *Handhabung* mehrerer Dateiformate mit ein. So können diverse Datenstrukturen, wie HTML-, Json-, Bild- oder Text-Dateien etc. dem KG zugeführt werden.

Die Verarbeitung der Daten wird in “Architecture of Knowledge Graph Construction Techniques” [9] in zwei Phasen unterteilt - in Wissens Extraktion und Wissens Fusion. Dabei werden in der Phase der Extraktion u.a. mit Techniken des maschinellen Lernens (ML) - wie Text Mining, NLP (Natural Language Processing) , - Entitäten herausgearbeitet. Die Entitäten können dann in der Fusions- Phase in Beziehung gebracht werden. Während dieser werden Relationen zwischen den Entitäten gesetzt, entweder durch eine eigene Ontologie bzw. Taxonomie oder mit Hilfe von Inferenzen von Drittanbietern wie Wikidata oder DBpedia. Die Konstruktion einer solchen Ontologie wird später in diesem Abschnitt noch eine nähere Betrachtung finden.

Die Aufgaben *extrahieren* und *fusionieren* können weitestgehend von Service Providern oder Frameworks übernommen werden (siehe Abbildung 2.8). Diese Funktionalität wird im Rahmen dieser Arbeit nicht näher erläutert, um den Fokus der Evaluierung des Einsatzes nicht zu verlieren. Sie können im erwähnten Paper “Architecture of Knowledge Graph Construction Techniques” [9] nachvollzogen werden.

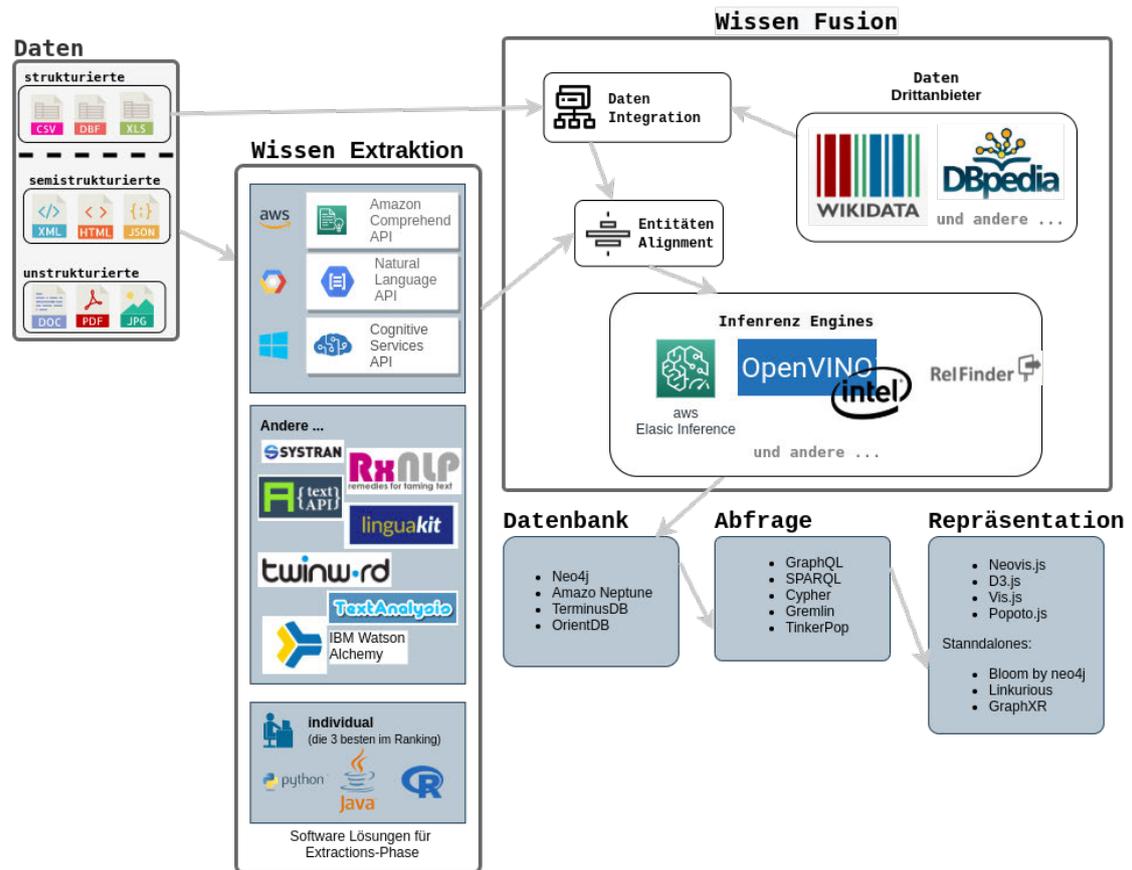


Abbildung 2.8: Software-(Teil-)Lösungen

Entscheidend für die Aufbereitung zum Speichern der Entitäten und Relationen ist die Art der Datenbank, welche im Abschnitt 2.4.4 spezifiziert werden.

Ontologie

Die Dissertation “Ontology-based modeling and configuration of construction processes using process patterns” zeigt unterschiedliche Modellierungsansätze, um eine Ontologie zu konstruieren [3]. Ein möglicher Ansatz besteht in der Adjazenzmatrix. Sie ermöglicht die Umsetzung eines gerichteten Graphen und eines Multigraphen.

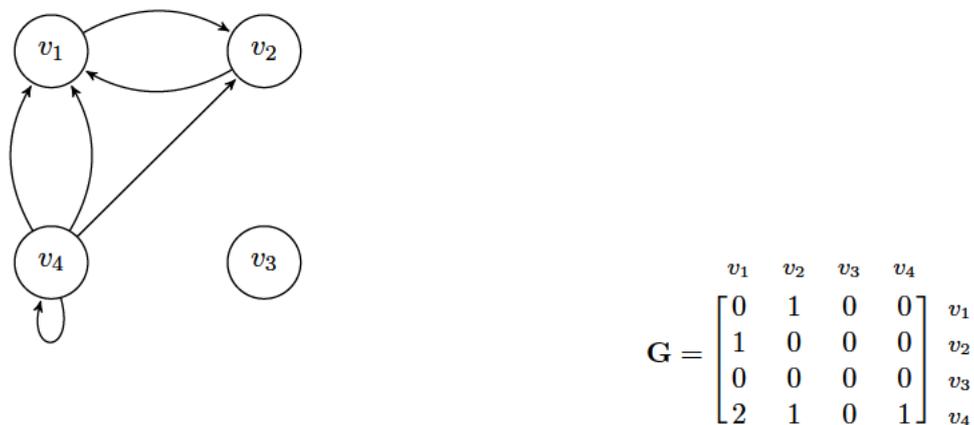


Abbildung 2.9: Graph und Adjazenzmatrix

2.4.4 Datenbank (Knowledge Graph)

Zur Speicherung der gewonnenen Wissensbasis (Ontologie) eignen sich unterschiedliche Datenbanksysteme. Die Wahl der Datenbank hängt von der Anzahl der Entitäten, Relationen und den Instanzen der Entitäten ab. Weist die Wissensbasis Ähnlichkeiten mit einem fachlichen Datenmodell auf, erweist sich die Anzahl der Entitäten als gering und die Anzahl der Instanzen als hoch, kann die Wahl auf eine relationale Datenbank fallen. Daher sind strukturierte Daten einer Relationalen Datenbank am einfachsten in einen KG einzubinden, da diese schon mit einer Fachlichen Logik aufbereitet sind.[5] Ist die Anzahl der Relationen hoch oder sogar generisch, eignet sich eine Graphendatenbank oder einem Triplestore - RDF (Resource Description Framework) Datenbank, eine der

Semantikregeln folgende Datenbank. Eine tiefer gehende Analyse, die sich dem Auswahlprozess der geeigneten Datenbanken widmet, ist im Rahmen dieser Arbeit nicht relevant. Der KG Prototyp dient lediglich der Evaluierung zur Darstellung von versteckten Beziehungen der Systemakten.

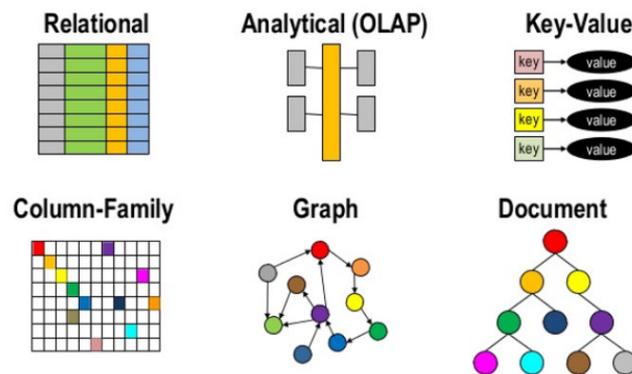


Abbildung 2.10: Datenbankstrukturen

2.4.5 Frontend (Visualisierung)

Die Visualisierung ist eine entscheidende Aufgabe für den KG. Sie stellt dem Nutzer die maschinell verknüpften Information bereit in Form eines Graphen bereit, der dann mit Hilfe von Oberflächen-Werkzeugen Antworten auf Fragen liefern soll.

2.4.6 Zusammenfassung

Aus dem Abschnitt KG lässt sich schließen, dass der KG aus Backend Frontend und einer Datenbank besteht. Der KG kann mit unterschiedlichen Dateiformaten umgehen. Das Backend extrahiert mithilfe von maschinellem Lernen logische Zusammenhänge zwischen den Datensätzen und speichert sie in einer ontologiebasierten Form ab. Zur Visualisierung des Gespeicherten Graphen sollte eine Schnittstelle die Graphdaten an das Frontend ausgeben. Die Visualisierung sollte Entitäten und deren Zusammenhänge klar darstellen, um dem Nutzer das Wissen zur Verfügung zu stellen - denn laut Definition gründet Wissen auf der Kenntnis der Beziehungen zwischen Konzepten/Informationen.

3 Analyse

Nachdem wichtige Grundlagen erläutert wurden, werden in diesem Kapitel die herausgearbeiteten Anforderungen an das in der Arbeit entwickelte KG-System vorgestellt.

3.1 Anforderungen

Filter Informationen

Nachdem der Graph geladen wurde, kann der User die Darstellung mit Werkzeugen wie Filtern und Suchen bearbeiten.

Suche: Nach der Eingabe eines Suchbegriffs werden Knoten mit den gesuchten Inhalten angezeigt. Neben den Ergebnissen der eigentlichen Suchanfrage, erscheinen weitere Teilgraphen, die ebenso interessante Nebeninformationen derselben Suchanfrage beinhalten.

Filter: Der User kann sich bestimmte Knoten-Kategorien anzeigen lassen, um die Komplexität der Relationen zu reduzieren und den Fokus auf ausgewählte Entitäten zu beschränken.

Arbeiten mit Informationen

Der User kann die Darstellung der Knoten und Kanten nach seinen Präferenzen verändern. Die *Graph-Darstellung* sollte in folgenden Werten variable einstellbar sein: display Name (Knoten/Kante), Größe (Knoten/Kante), Position (Graph/Elemente), selektieren von Knoten/Kante, verstecken von Knoten, pinen von Knoten. Die flexible Darstellung soll dem Nutzer eine gute Perspektive auf das Wissen ermöglichen, z.B. um Detailwissen effizient greifbar zu machen.

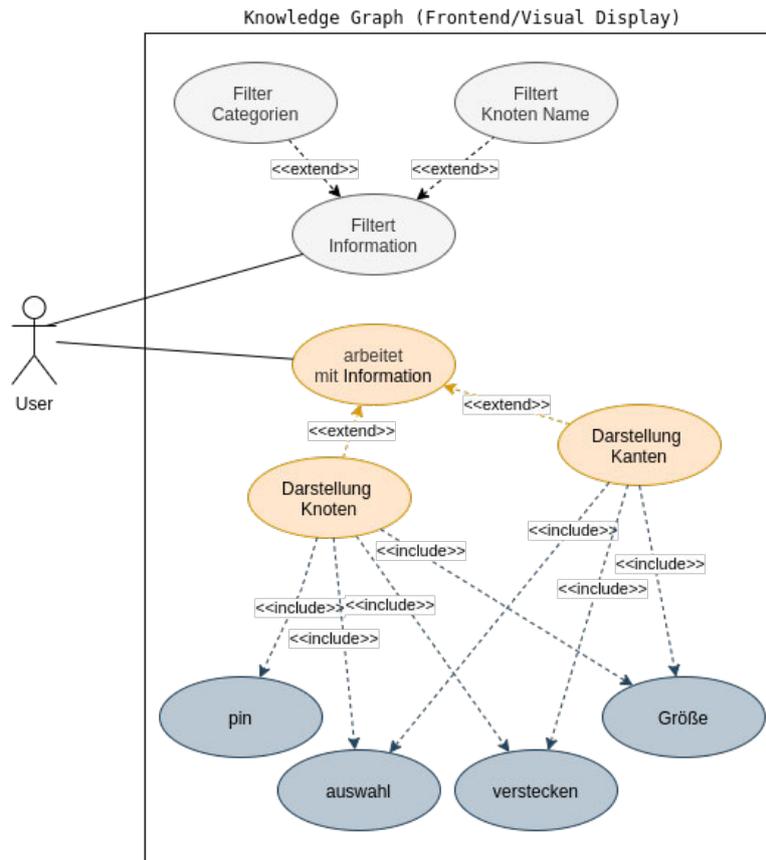


Abbildung 3.1: Anforderungen an das System

3.2 Sichten

Wie in den Grundlagen umrissen, kann ein KG nach dem Prinzip der Trennung der Verantwortlichkeiten (“Separation of concerns”) in Backend und Frontend geplant werden. Des Weiteren werden die Aufgaben Extraktion, Fusion, Search und UI in den jeweiligen Modulen umgesetzt.

3.2.1 Komponenten und Module

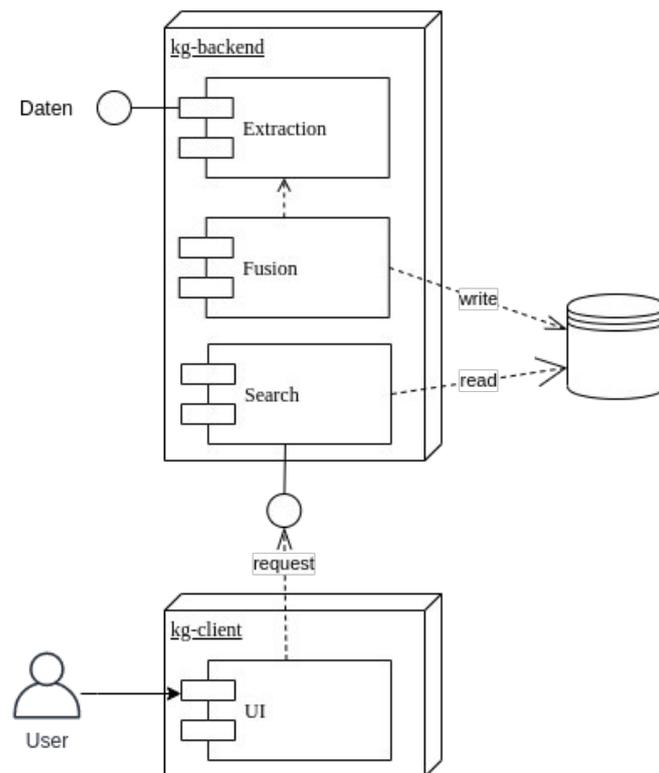


Abbildung 3.2: Komponenten und Module

3.2.2 Module und Klassen

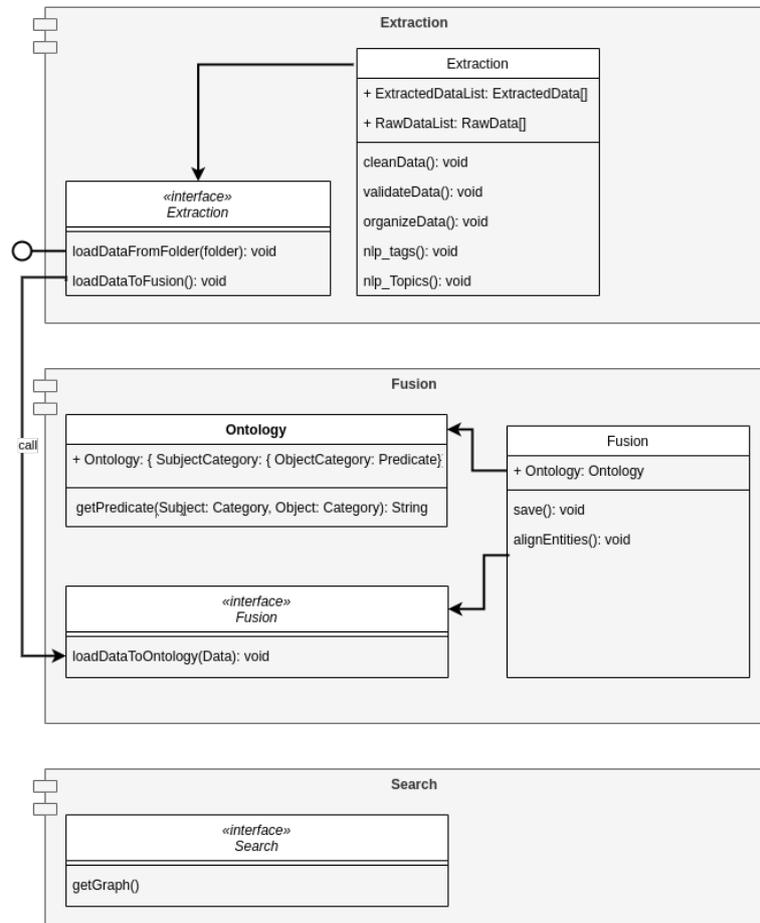


Abbildung 3.3: Module und Klassen

3.3 API-Spezifikation und Datenmodelle

Um die Rest- Schnittstelle des Backends aktuell zu halten, wurde das Werkzeug Swagger eingesetzt. Eine graphische Aufbereitung der API Spezifikation und die Dokumentation der Datenmodelle sind im Anhang A.1 zu finden.

3.4 Ontologie

Erwartete Entitäten und Relationen werden in folgender Ontologie angelegt (siehe Abbildung 3.4). Es ist zu beachten, dass der Graph gerichtet ist.

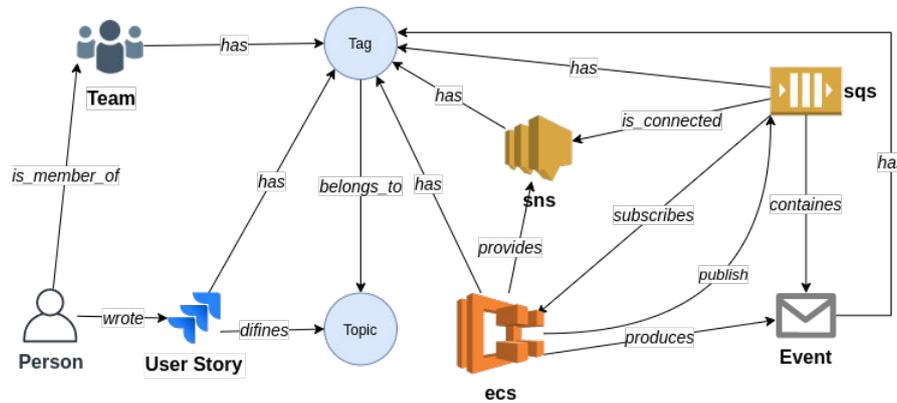


Abbildung 3.4: Erwartete Ontologie mit zusätzlichen experimentellen Entitäten

Eine generische $N \times N$ Matrix (Abbildung 3.5) für die Ontologie gewährleistet das Hinzufügen von unbekanntem Entitäten und Relationen.

3 Analyse

	tag	story	person	team	sns	sqs	event	ecs	text	owner	lambda	role	category	project
tag	tag_tag	tag_story	tag_person	tag_team	tag_sns	tag_sqs	tag_event	tag_ecs	tag_text	tag_owner	tag_lambda	tag_role	tag_category	tag_project
story	has	story_story	is_written_by	story_team	story_sns	story_sqs	story_event	story_ecs	story_text	story_owner	story_lambda	story_role	story_category	story_project
person	person_tag	person_story	person_person	is_member_of	person_sns	person_sqs	person_event	person_ecs	person_text	person_owner	person_lambda	person_role	person_category	person_project
team	has	team_story	team_person	team_team	team_sns	team_sqs	team_event	team_ecs	team_text	team_owner	team_lambda	team_role	team_category	team_project
sns	has	sns_story	sns_person	sns_team	sns_sns	sns_sqs	sns_event	sns_ecs	sns_text	sns_owner	sns_lambda	sns_role	sns_category	sns_project
sqs	has	sqs_story	sqs_person	sqs_team	is_connected	sqs_sqs	contains	subscribe_ecs	sqs_text	sqs_owner	sqs_lambda	sqs_role	sqs_category	sqs_project
event	event_tag	event_story	event_person	event_team	event_sns	event_sqs	event_event	event_ecs	event_text	event_owner	event_lambda	event_role	event_category	event_project
ecs	has	ecs_story	ecs_person	ecs_team	publish_sns	publish_sqs	produce	ecs_ecs	ecs_text	ecs_owner	ecs_lambda	ecs_role	ecs_category	ecs_project
text	text_tag	text_story	text_person	text_team	text_sns	text_sqs	text_event	text_ecs	text_text	text_owner	text_lambda	text_role	text_category	text_project
owner	owner_tag	owner_story	owner_person	owner_team	owner_sns	owner_sqs	owner_event	owner_ecs	owner_text	owner_owner	owner_lambda	owner_role	owner_category	owner_project
lambda	lambda_tag	lambda_story	lambda_person	lambda_team	lambda_sns	lambda_sqs	lambda_event	lambda_ecs	lambda_text	lambda_owner	lambda_lambda	lambda_role	lambda_category	lambda_project
role	role_tag	role_story	role_person	role_team	role_sns	role_sqs	role_event	role_ecs	role_text	role_owner	role_lambda	role_role	role_category	role_project
category	category_tag	category_story	category_person	category_team	category_sns	category_sqs	category_event	category_ecs	category_text	category_owner	category_lambda	category_role	category_category	category_project
project	project_tag	project_story	project_person	project_team	project_sns	project_sqs	project_event	project_ecs	project_text	project_owner	project_lambda	project_role	project_category	project_project

how to read the table:

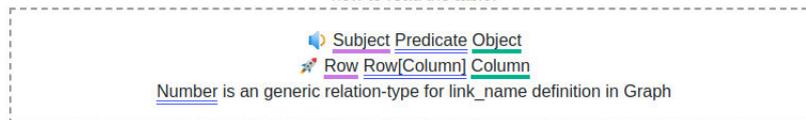


Abbildung 3.5: Generierte Matrix mit default Ontologie

4 Umsetzung

Dieses Kapitel befasst sich mit der Implementierung eines KG-Prototyps. Dabei sollen nur wenige Besonderheiten wie die Konstruktion und Speicherung der Ontologie (Abschnitt 4.1).

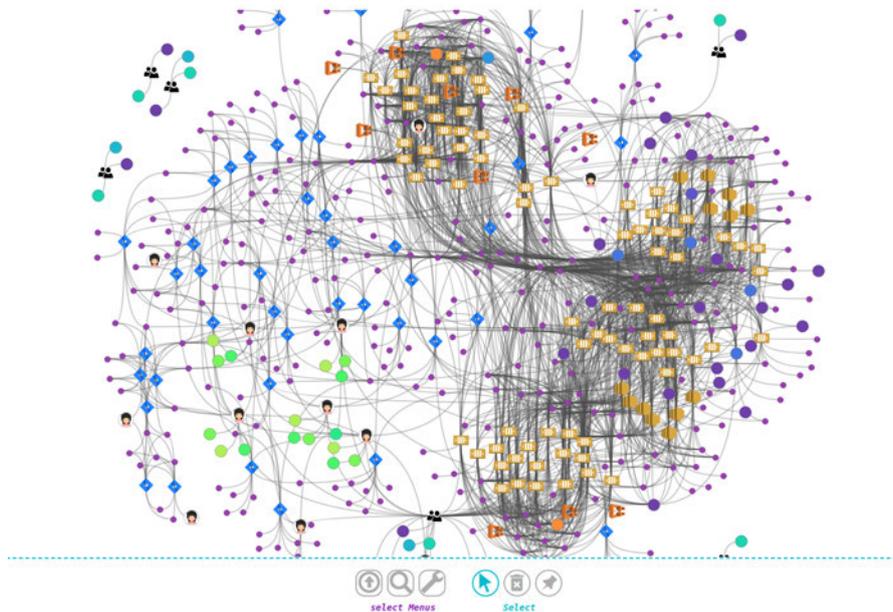


Abbildung 4.1: Implementierte Benutzer-Oberfläche zur Visualisierung des Knowledge Graphen

4.1 Ontologie

4.1.1 Implementierung

Während der Implementierung änderten sich regelmäßig die Namen der Kanten zwischen den Knoten. So wurde eine SVG Datenstruktur (Abbildung 4.2) eingesetzt um das Diagramm (die Ontologie) mit der Implementierung konsistent zu halten.¹

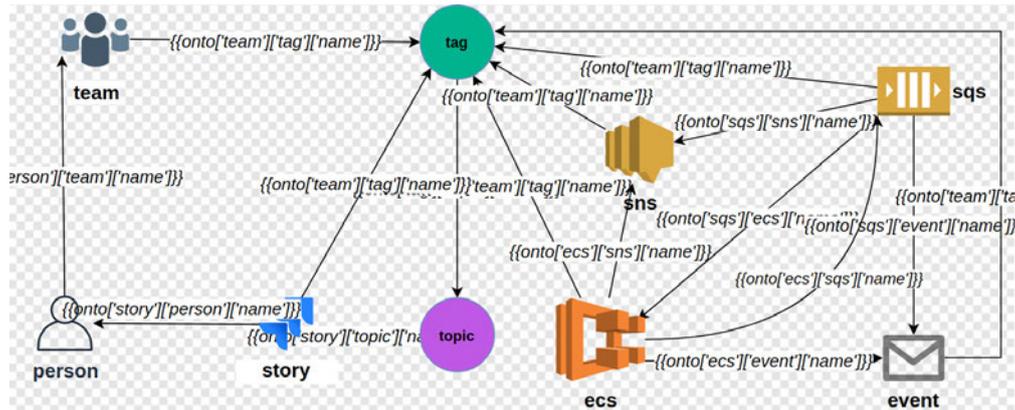


Abbildung 4.2: Generische Lösung für die Benennung der Relationen (Prädikate)

Logik

Die implementierte Ontologie ist nicht starr, sie kann dynamisch mit weiteren Entitäten und generischen Relationen angereichert werden. Eine zukünftiges Feature könnte den Anwendern ermöglichen, erkannte neue Relationen zu validieren und zu benennen.

Listing 4.1: Ontologie Implementierung

```
class Ontology(object):

    default_predicates = {} # provide some default realltion between
                           # the Categories (Entities)
                           # Setup in subsection predicates

    def __init__(self, categories: List[str]):
        self.ontoloy = {}
        for category in categories:
```

¹Implementierung befindet sich im Anhang A.3.

```
self.ontoloy[category] = {}

for subject, objects in self.ontoloy.items():
    for category in categories:
        self.__add_predicate(subject, subject, category)
        objects[category] = {'name': subject+'_'+category}

for subject, objects in self.default_predicates.items():
    for obj in objects:
        self.__add_predicate(subject, obj[0], obj[1], self.COLOR )

def __add_predicate(self, subj: str, predicate: str, obj: str, default:str
=''):
    self.ontoloy[subj][obj] = {'name': predicate, 'color': default}

def get_onto(self) -> dict:
    return self.ontoloy

def get_predicate(self, subj: str, obj: str):
    return self.ontoloy[subj][obj]
```

Default Prädikate

Nachdem die Extraktion eine Liste mit Entitäten bereitstellt, kann mit Hilfe einer Ontologie (siehe Analyse Ontologie 3.4) ein Graph gezeichnet werden. Die Ontologie folgt dann den Gesetzen der Semantik, und stellt Prädikate zwischen Subjekt und Objekt her.

Listing 4.2: Default Prädikate der These

```
default_predicates = {
    'person':
        [
            ['is_member_of', 'team']
        ],
    'team':
        [
            ['has', 'tag']
        ],
}
```

```
'tag':
  [
    ['belongs_to', 'topic']
  ],
'story':
  [
    ['is_written_by', 'person'],
    ['under', 'topic'],
    ['has', 'tag']
  ],
'ecs':
  [
    ['publish_sns', 'sns'],
    ['produce', 'event'],
    ['publish_sqs', 'sqs'],
    ['has', 'tag']
  ],
'sqs':
  [
    ['subscribe_ecs', 'ecs'],
    ['contains', 'event'],
    ['is_connected', 'sns'],
    ['has', 'tag']
  ],
'sns':
  [
    ['has', 'tag']
  ],
'sns':
  [
    ['has', 'tag']
  ]
}
```

4.1.2 Herausforderung

Generisches Hinzufügen von neuen Kategorien der Knoten und Kanten wurde hier mit einer Adjazenzmatrix bewerkstelligt; Diese Lösung kann beliebig um Knoten und Kan-

ten erweitert werden und bleibt in ihrer Performance der Operationen linear mit den Relationen.

5 Tests

Laut des Anforderungen-Katalogs sollen unterschiedliche Funktionalitäten bereit gestellt werden. Diese sollen im Folgenden in visueller Form belegt werden.

Test - Prozessieren unterschiedlicher Datenquellen

Es können .json .csv und .txt Dateien geladen werden. Die Validierung wird im Backend durch Wordclouds nachempfunden, siehe Anhang A.4. Die Abbildung 5.1 zeigt einen Ausschnitt vom Backend während des Arbeitsschritts Extraktion.

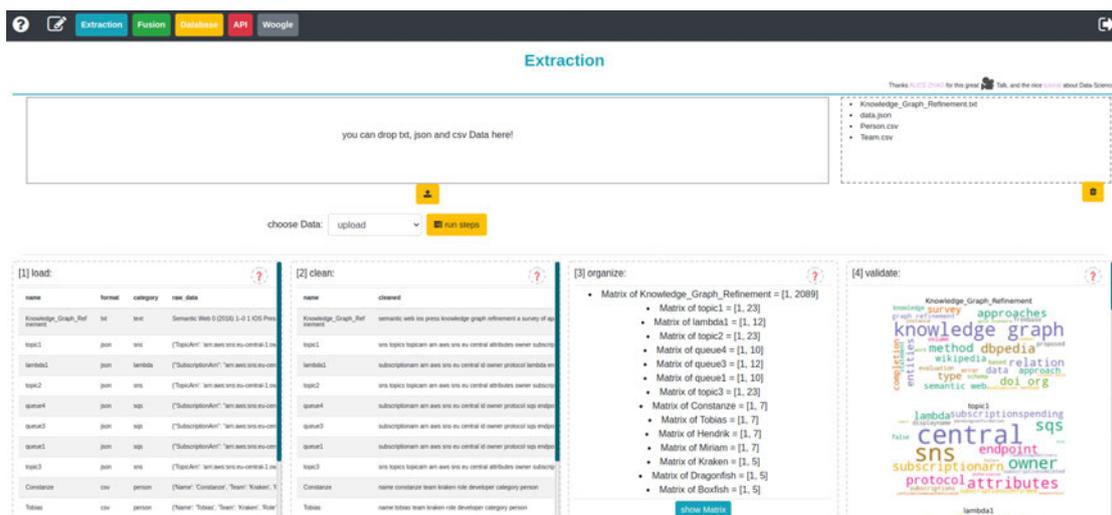


Abbildung 5.1: Test - Händeln unterschiedlicher Dateiformate und Extraktionsschritt

Test - Darstellung Kopplung der Software-Ressourcen von zwei Teams

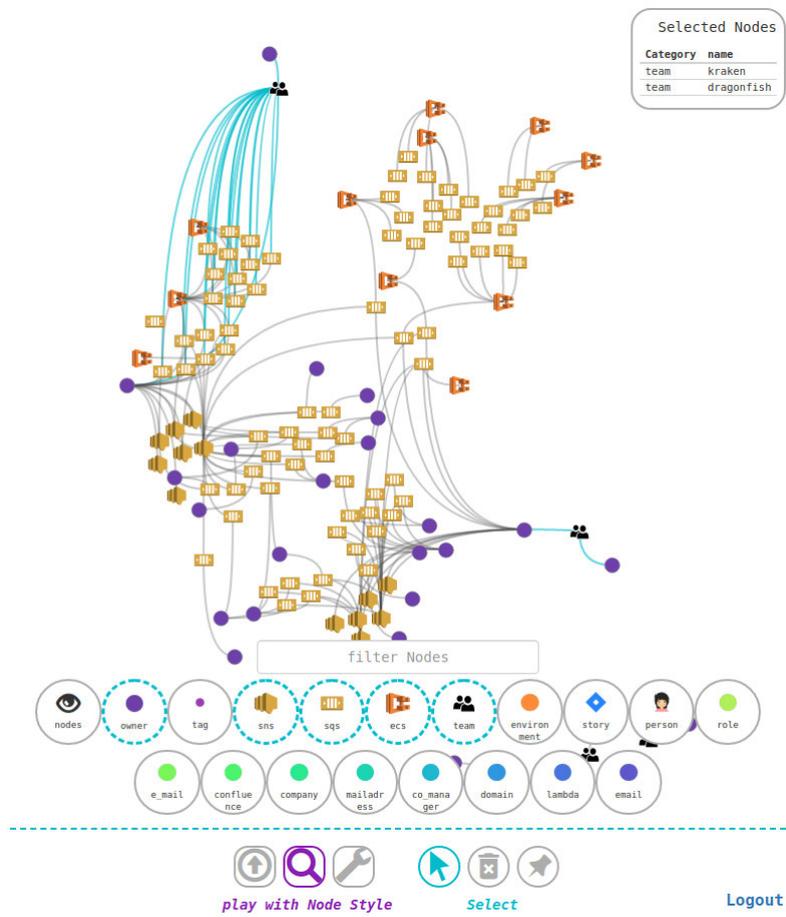


Abbildung 5.2: Test - Verbindung von Team Kraken und Team Dragonfish über ihre Software-Ressourcen

Test - Informationen Filtern

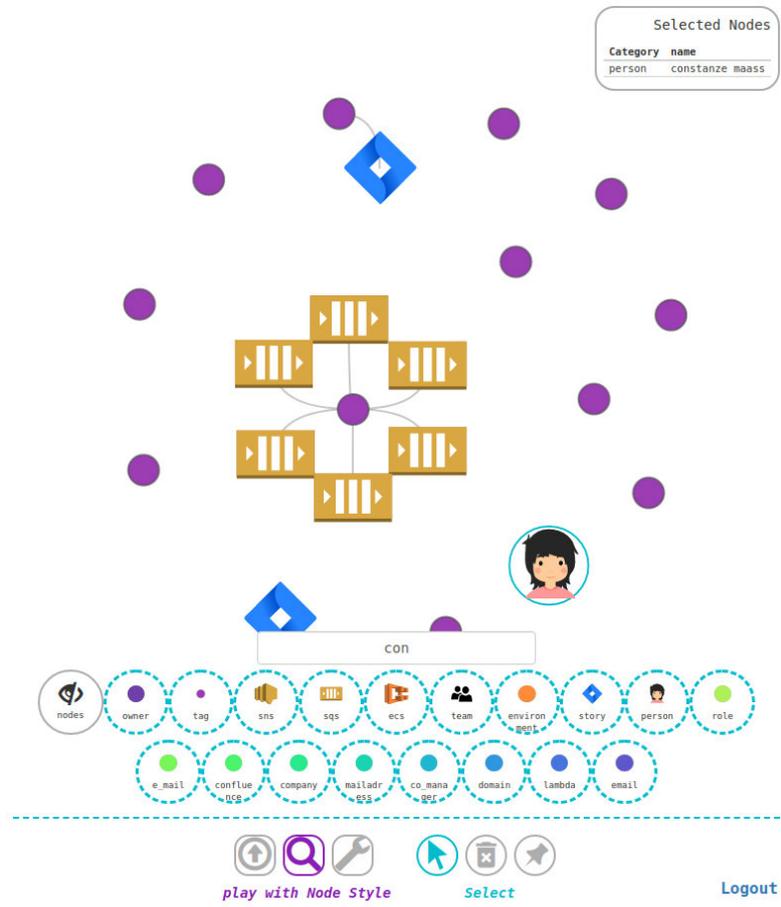
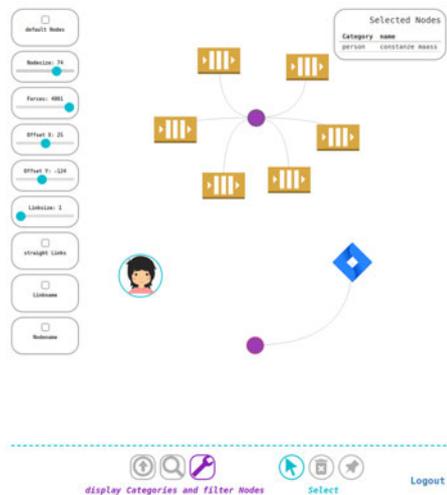
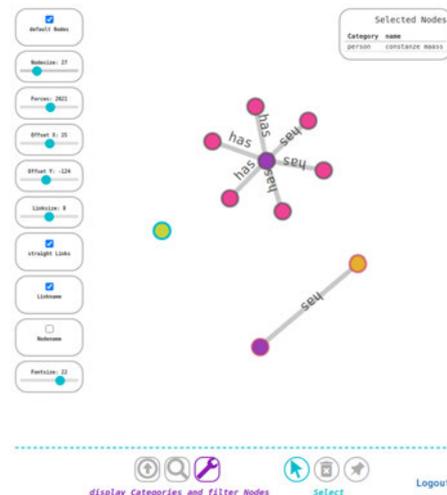


Abbildung 5.3: Anforderung- Graph filtern mit bestimmten Begriffen

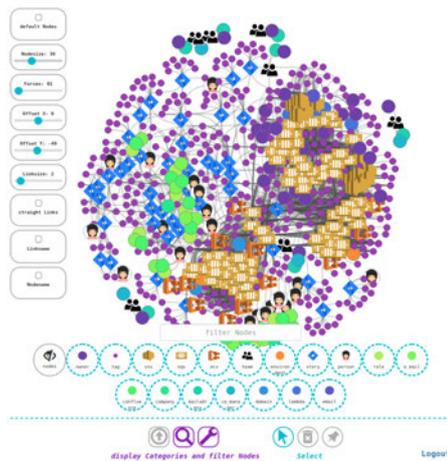
Test - Darstellung bearbeiten



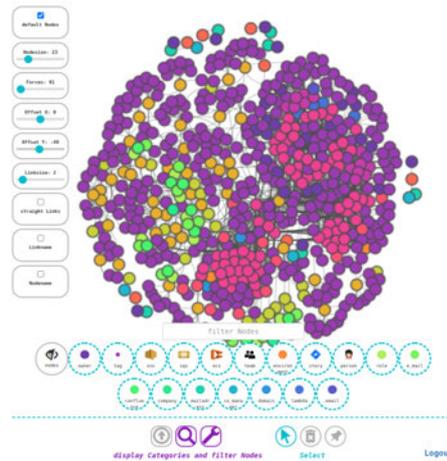
(a) Einstellungen: (curved links) (Icon-Nodes) (nodesize:74)



(b) bearbeitet Einstellungen: (straight links) (default-Nodes) (linknames display) (nodesize:27) (linksize:8)



(c) bearbeitet Einstellungen: Ansicht mit vordefinierten Ontologie Icons



(d) Einstellungen: Default Knoten

Abbildung 5.4: Anforderung- Graph-Darstellung ändern

Zusammenfassung

Die hier aufgeführten Tests fügen sich in den Rahmen der Arbeit ein. Sie validieren hier nur die User-Anforderung an die Software. Sollte die Software in Produktion gehen, ist auch eine umfassendere Testumgebung bereitzustellen.

6 Evaluierung

Die in den Grundlagen heraus gearbeiteten Faktoren für das Gelingen von KM, wurden in eine Umfrage¹ an zwei agile Teams weitergereicht. Im Folgenden werden die Ergebnisse der Umfrage abschnittsweise betrachtet und anhand der Kriterien für Knowledge Management bewertet.

6.1 Auswertung der Umfrage

erwartete Teilnehmer: ca 30 (Anzahl zwei Teams)

reale Teilnehmer: 4

relative Teilnahme: 13%

Die geringe Teilnahme (13%) an der Bewertung des KMS lässt sich auf unzureichende Rahmenbedingungen zurückführen. Der zeitliche Aspekt könnte für die Test- Teams eine große Rolle spielen: Steht ein Raum für Umfragen und Tests dieser Art bereit? Können die Teammitglieder Zeit dafür erübrigen? Eine Aufgabe der Organisation besteht darin, diese Rahmenbedingungen für das Gelingen von KM zu gewährleisten.

6.1.1 Einführung

Ein Einführungsvideo und die Demonstration des Systems im zweiten Video sollen die Vision von KM verdeutlichen und den Wille zu KM wecken, gemäß der geforderten Rahmenbedingungen der KM.

Ergebnisse

¹Siehe Anhang A

Anmerkung	Person-A	Person-D
Einführung	Like it sooooo much ... still!	Ich finde die Idee eines Knowledge Graph sehr hilfreich. Momentan sind die Information wie im Video beschrieben in diversen Tools zu finden.
Demo	Schöne Demo! Gefällt mir ...	Woogle wirkt zwar auf den ersten Blick wie ein Wollknäuel, durch die Tools/ Filter und unterschiedliche Symbole ist es gut struktuiert. Ich kann mir vorstellen, dass es nach einer ersten Eingewöhnungszeit sehr gut und intuitiv nutzbar ist. Für das IT-Management macht wahrscheinlich eine kurze Schulung Sinn, damit sie wissen, wie es genutzt werden kann. Ein paar Fragen: Könnte man sich in späteren Versionen eigene Filter/ Sichten abspeichern? Wo sind die Daten abgelegt? Ist es datenschutzsicher? Insgesamt eine tolle Sache und ich könnte mir vorstellen, dass es nach einer ersten Testphase auch von anderen Teams/ projektweit eingesetzt werden kann. Das wird dann performance-seitig wahrscheinlich eine Herausforderung, aber wenn es funktioniert wäre es eine gute Dokumentation

Tabelle 6.1: Anmerkungen zur Motivation

Auswertung

50% - 75% der Teilnehmenden haben Anmerkungen hinterlegt, das wird hier dem Wille und Interesse an der KM unterstellt. Ein positiv zu bewertender Aspekt.

6.1.2 Praxis

Ein Einführungsvideo und die Demonstration des Systems im zweiten Video sollen die Vision von KM verdeutlichen und den Wille zu KM wecken, wie die Rahmenbedingungen der KM es fordern.

Ergebnisse

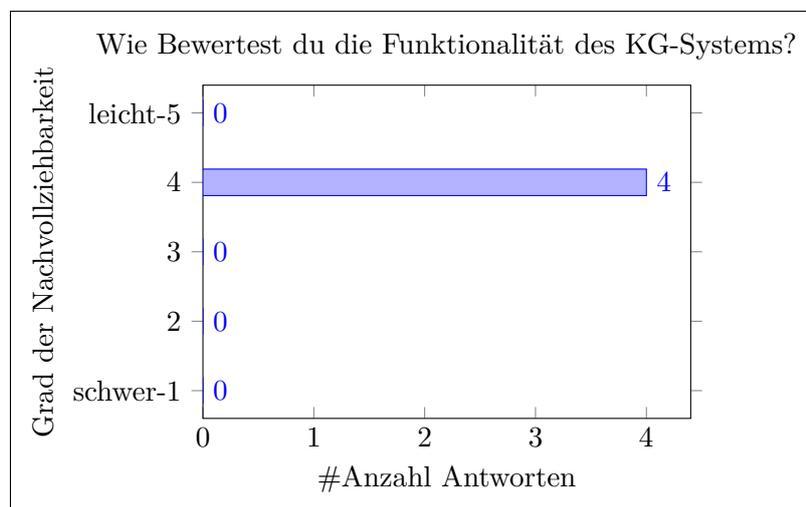


Abbildung 6.1: Bewertung der Funktionalität des KG-Systems

Anmerkung	Person-A	Person-D
Funktionalität	SNS und SQS sind als spezifische Filtermöglichkeiten vorhanden ... kann man sich über UI-Filter auch spezielle Tags anzeigen lassen?	Eine Mini-Schulung wäre hilfreich, um alle Funktionalitäten und deren Nutzen zu verstehen. Es wirkt auf den ersten Blick aber sehr intuitiv

Tabelle 6.2: Anmerkungen zur Einleitung

Auswertung

Die Teilnehmenden bewerten das System als intuitiv und weitgehend leicht verständlich (siehe Diagramm 6.1)). Auch der Hinweis für Erweiterungen ist als Interesse und Klarheit über die möglichen Vorteile des Einsatzes zu bewerten (siehe Tabelle Anmerkung Person A 6.2).

6.1.3 Bewertung

Die Teilnehmenden sollten den Einsatz mit Hinblick auf die Ziele von KM bewerten. Dabei wurden Ziele gewählt, die im Aktionsradius vom Team liegen. Die persönliche Bewertung des KG ist abschließend eine Einschätzung der intrinsischen Motivation.

Ergebnisse

Die Abbildungen 6.2 und 6.3 sind eine graphische Darstellung der Ergebnisse aus den Tabellen A.1 und A.2².

²Siehe Anhang.

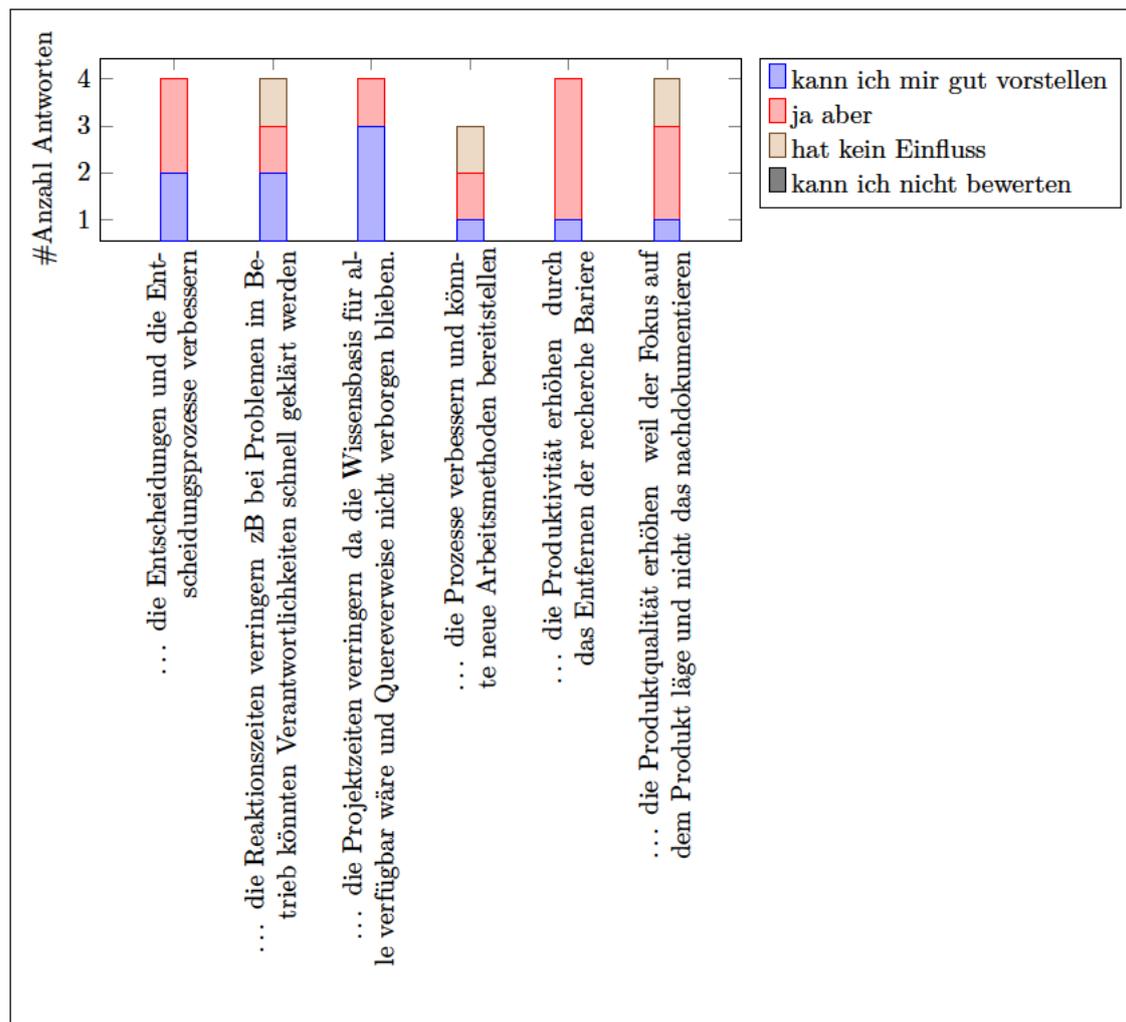


Abbildung 6.2: Der Einsatz eines Knowledge Graphen im Wissensmanagement könnte ...

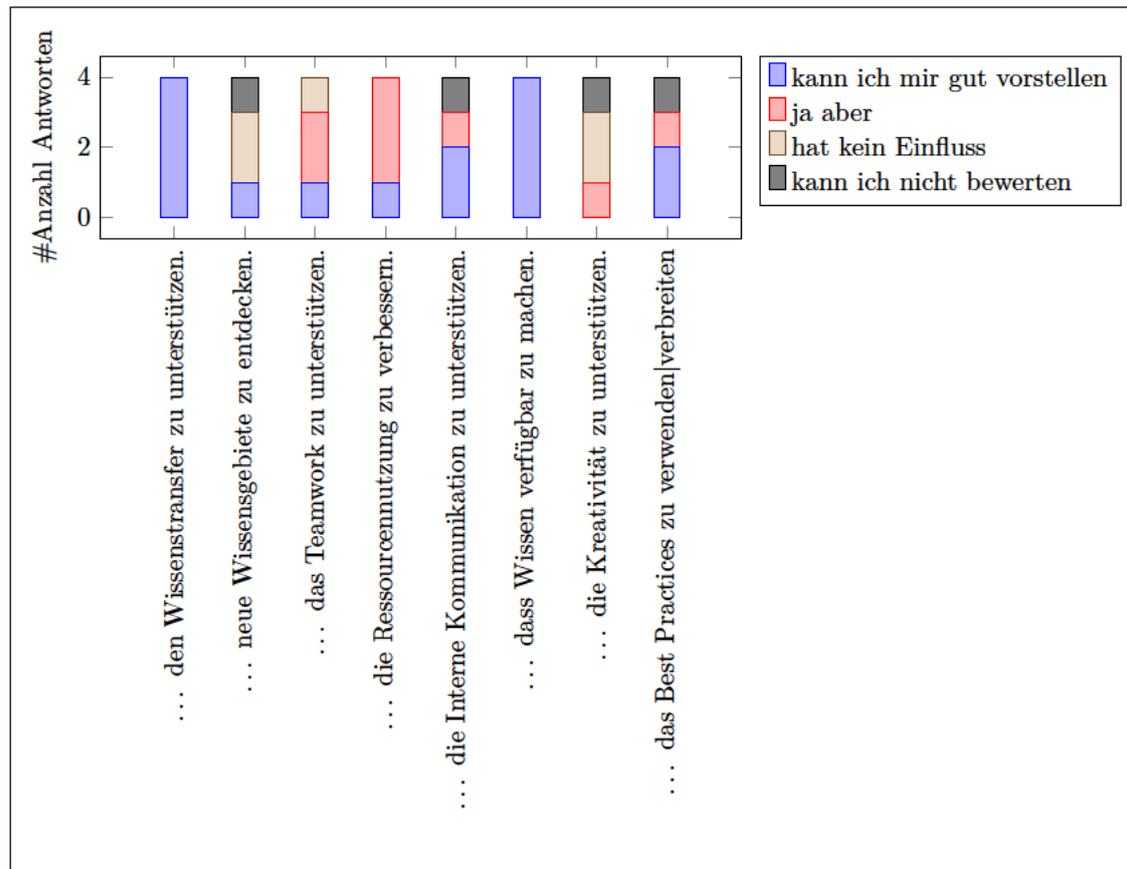


Abbildung 6.3: Mit dem Einsatz eines Knowledge Graphen im Wissensmanagement ist es möglich ...

Auswertung

Die Ziele und der Nutzen von KM wird weitestgehend erkannt und verstanden. Die persönliche Bewertung signalisiert eine positive intrinsische Motivation.

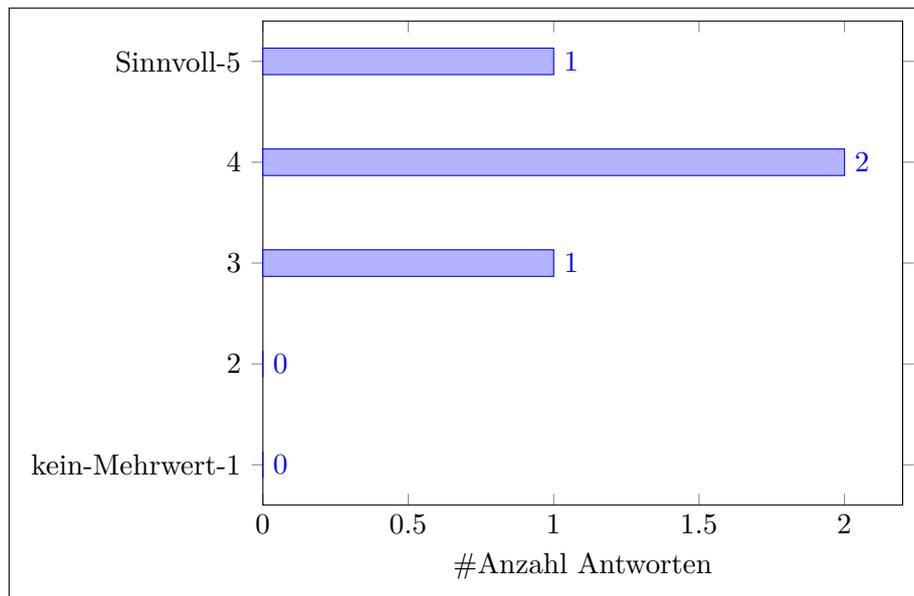


Abbildung 6.4: Wie bewertest du den Einsatz eines Knowledge Graph Systems in der Dokumentation (Projekt- / System- / ect-) ,

6.2 Zusammenfassung

Die Teilnehmer bewerten die KG-Technologie von mittelmäßig bis sehr gut. Zu beachten ist, dass der Prototyp keinesfalls alle Funktionen eines vollständigen KMS bietet. So könnten die mittelmäßigen Bewertungen eine Optimierung erfahren, wenn sich das System in seiner weiteren Entwicklung den Standardfunktionen von KM annähern würde. Die geringe Teilnahme ist nicht als Interesselosigkeit zu bewerten, sondern als Aufforderung an die Organisation, den Teams den Rahmen und die Zeit zur kreativen Weiterbildung einzuräumen.

7 Fazit

Die Annahme, dass man eine automatisierte Dokumentation von teamübergreifenden Systemen erstellen kann, ist mit der verknüpften Darstellung der beiden Teams mit ihren Softwaresystemen, belegt. Doch da nur ein kleiner Teil des System-Wissens, in dem begrenzten Rahmen einer Bachelor-Arbeit dargestellt werden, lässt sich abschließend noch keine konkrete Aussage zum Einsatz der Knowledge Graph Technologie machen. Weiterführend könnten Definitionen von Wissensbasis Providern, wie Wikidata automatisiert hinzugefügt werden. Oder User Stories mit der implementierten Software in Verbindung gebracht werden, was in experimenteller Form schon teilweise umgesetzt wurde.

Die Herausforderung besteht immer noch darin, unbeaufsichtigt Wissen aus den Informationen zu produzieren. Die Erstellung einer umfassenden Unternehmens-Ontologie ist ein Schlüsselfaktor für die Gewinnung valider Graphen. Eine Grundvoraussetzung für Implementierung einer solchen Software liegt in den unternehmerisch bereit gestellten Geld- und Zeittressourcen.

Ein genau auf die Anforderungen dieser Arbeit passendes Tool (siehe Abbildung 7.1), um das Wissen in einem Unternehmen sichtbar/verfügbar zu machen, wurde bereits entwickelt. Die Firma GraphAware stellt seit dem 12. April 2021 eine erste Demoversion von HumeHume¹ bereit. Es bietet sich eine Evaluierung, für den Einsatz einer solchen Dritt-Anbieter-Lösung an

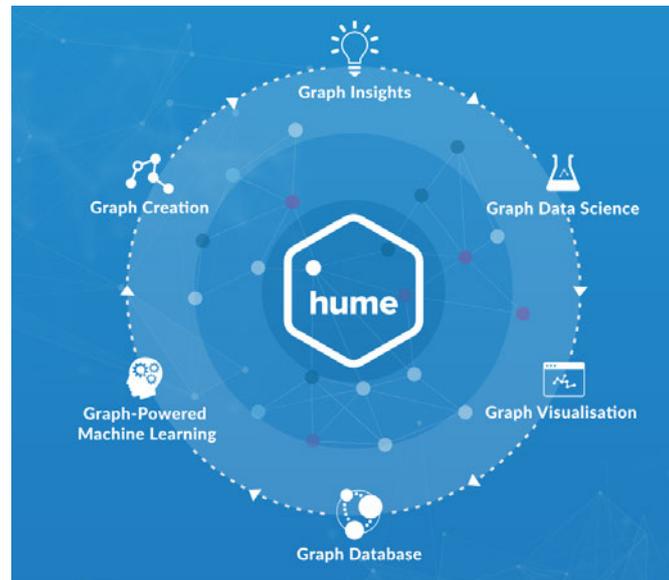


Abbildung 7.1: Hume Technologie

Eine KM mit einem möglichst kleinem Regelwerk zur Erweiterung der Wissensbestände, könnte wohl eine größere Wissensbasis schaffen als eine KM mit vielen starren Regeln. Die damit verbundene niedrigere Hemmschwelle, das eigene Wissen zu teilen, schafft sicherlich einen hohen Anreiz dafür, eine größere Menge an Informationen in das System einzuspeisen. Die Verknüpfung der Informationen ist dann Aufgabe des Knowledge-Graphen. Das System erschafft so Brücken zum Wissen.

“Wissen ist Macht – Macht ist Wissen”²

¹ <https://www.graphaware.com/products/hume/>

² Wilhelm Liebknecht in einem Vortrag.

Literaturverzeichnis

- [1] Agiles Manifest. (2021). – URL <https://agilemanifesto.org/iso/de/manifesto.html>
- [2] IT-Grundschutz-Bausteine. (2021). – URL https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/IT-Grundschutz-Kompodium/IT-Grundschutz-Bausteine/Bausteine_Download_Edition_node.html
- [3] BENEVOLENSKIY, Alexander: *Ontology-based modeling and configuration of construction processes using process patterns = Ontologie-basierte Modellierung und Konfiguration der Bauprozesse mit Hilfe von Prozessvorlagen*. Institut für Bauinformatik, Fakultät Bauingenieurwesen, TU Dresden, 2016
- [4] HÜTTENEGGER, Georg: *Open Source Knowledge Management mit 9 Tabellen*. Springer-Verlag Berlin Heidelberg, 2006 (Xpert.press). – URL <https://external.dandelon.com/download/attachments/dandelon/ids/DEAGI321697E2DA42DD28C12571140070F571.pdf>
- [5] LEPENDU, Dou D.: Using ontology databases for scalable query answering. In: *Journal of intelligent information systems*, 37(2) (2011), S. 217–244. – URL <https://doi.org/10.1007/s10844-010-0133-4>
- [6] REISS, Manuela ; REISS, Georg: *Praxisbuch IT-Dokumentation Betriebshandbuch, Systemdokumentation und Notfallhandbuch im Griff*. [Neue Aufl.]. Hanser, 2019. – URL <https://external.dandelon.com/download/attachments/dandelon/ids/DE00187C3244BA0135E14C1257C5800412422.pdf>
- [7] SICULAR, Svetlana ; VASHISTH, Shubhangi: Hype Cycle for Artificial Intelligence. In: *Gartner* (2020). – URL <https://www.gartner.com/en/documents/3988006>

- [8] WOLFF, Eberhard: *Microservices Grundlagen flexibler Softwarearchitekturen*. 2., aktualisierte Auflage. dpunkt.verlag, 2018. – URL <http://d-nb.info/1160503982/04>
- [9] ZHAO, Zhanfang ; HAN, Sung-Kook ; SO, In-Mi: Architecture of Knowledge Graph Construction Techniques. In: *International Journal of Pure and Applied Mathematics* (2018), 2. – URL <http://www.ijpam.eu>. – ISSN 1869-1883-1314-3395

A Anhang

A.1 API-Spezifikation und Datenmodelle

Listing A.1: Rest Api Dokumentation

```
openapi: 3.0.1
info:
  title: Knowledge Graph Backend
  description: this is the api spezifikation for Knowledge Graph Engine
  termsOfService: https://bachelor.wusel.space/
  contact:
    email: cosmaastani@googlemail.com
  version: 1.0.0
servers:
  - url: https://bachelor.wusel.space/v1
tags:
  - name: Search
    description: API for Search Requests from Knowledge - Graph Client
paths:
  /graph:
    get:
      tags:
        - Search
      summary: Returns a list of all Nodes and Links
      operationId: get_graph
      responses:
        "200":
          description: successful operation
          content:

            application/json:
              schema:
                type: object
```

```

        properties:
          links:
            type: array
            items:
              $ref: '#/components/schemas/Link'
          nodes:
            type: array
            items:
              $ref: '#/components/schemas/Node'
    "404":
      description: graph not found
      content: {}
    x-swagger-router-controller: kg_backend.search.controller
    x-openapi-router-controller: kg_backend.search.controller
components:
  schemas:
    Node:
      type: object
      properties:
        id:
          type: integer
          format: int64
        name:
          type: string
          example: from_to_service_return_queue
        category:
          type: string
          example: person
        data_format:
          type: string
          default: txt
        data:
          type: string
          example: raw data as string
      description: Node Schema for Graph UI
      example:
        name: from_to_service_return_queue
        id: 6
        category: story
        data: raw data as string

```

```
    data_format: txt
Link:
  type: object
  properties:
    id:
      type: integer
      format: int64
    name:
      type: string
      example: return_queue
    sid:
      type: integer
      format: int64
    tid:
      type: integer
      format: int64
  description: Link Schema for Graph UI
  example:
    name: return_queue
    id: 0
    sid: 5
    tid: 3
```

Knowledge Graph Backend 1.0.0 OAS3

[/v1/openapi.json](#)

this is the api spezification for Knowledge Graph Engine

[Terms of service](#)

[Contact the developer](#)

Servers

/v1

Search API for Search Requests from Knowledge - Graph Client

GET /graph Returns a list of all Nodes and Links

Try it out

Parameters

No parameters

Responses

Code	Description	Links
200	successful operation	No links

Media type

application/json

Controls Accept header.

Example Value | Schema

```

{
  "links": [
    {
      "id": 0,
      "name": "return_queue",
      "sid": 5,
      "tid": 3
    }
  ],
  "nodes": [
    {
      "category": "story",
      "data": "raw data as string",
      "data_format": "txt",
      "id": 6,
      "name": "from_to_service_return_queue"
    }
  ]
}

```

404	graph not found	No links
-----	-----------------	----------

Media type

Schemas



```
Link ▾ {
  description:      Link Schema for Graph UI

  id                integer($int64)
  name              string
                   example: return_queue
  sid               integer($int64)
  tid               integer($int64)
}
```

```
Node ▾ {
  description:      Node Schema for Graph UI

  category          string
                   example: person
  data              string
                   example: raw data as string
  data_format       string
                   default: txt
  id                integer($int64)
  name              string
                   example: from_to_service_return_queue
}
```

A.2 Tabellen

Mit dem Einsatz eines Knowledge Graphen im Wissensmanagement ist es möglich ...

Aussage	Person-A	Person-B	Person-C	Person-D
... das Teamwork zu unterstützen.	ja, aber	ja, aber	kann ich mir gut vorstellen	hat kein Einfluss
... dass Wissen verfügbar zu machen.	kann ich mir gut vorstellen			
... den Wissenstransfer zu unterstützen.	kann ich mir gut vorstellen			
... die Interne Kommunikation zu unterstützen.	ja, aber	kann ich mir gut vorstellen	kann ich mir gut vorstellen	kann ich nicht bewerten
... die Kreativität zu unterstützen.	hat kein Einfluss	ja, aber	hat kein Einfluss	kann ich nicht bewerten
... die Ressourcennutzung zu verbessern.	ja, aber	kann ich mir gut vorstellen	ja, aber	ja, aber
... neue Wissensgebiete zu entdecken.	hat kein Einfluss	kann ich mir gut vorstellen	hat kein Einfluss	kann ich nicht bewerten

Tabelle A.1: Ergebnis Umfrage (Bewertung-Aussagen I)

Der Einsatz eines Knowledge Graphen im Wissensmanagement könnte ...				
Aussage	Person-A	Person-B	Person-C	Person-D
... die Reaktionszeiten verringern,zB bei Problemen im Betrieb könnten Verantwortlichkeiten schnell geklärt werden.	kann ich mir gut vorstellen	ja, aber	hat kein Einfluss	kann ich mir gut vorstellen
... die Entscheidungen und die Entscheidungsprozesse verbessern	ja, aber	kann ich mir gut vorstellen	kann ich mir gut vorstellen	ja, aber
... die Produktivität erhöhen , durch das Entfernen der recherche Barriere.	ja, aber	ja, aber	ja, aber	kann ich mir gut vorstellen
... die Produktqualität erhöhen, weil der Fokus auf dem Produkt läge und nicht das 'Nachdokumentieren'.	ja, aber	hat kein Einfluss	ja, aber	kann ich mir gut vorstellen
... die Projektzeiten verringern, da die Wissensbasis für alle verfügbar wäre und Quereverweise nicht verborgen blieben.	kann ich mir gut vorstellen	ja, aber	kann ich mir gut vorstellen	kann ich mir gut vorstellen
... die Prozesse verbessern und könnte neue Arbeitsmethoden bereitstellen!		ja, aber	hat kein Einfluss	kann ich mir gut vorstellen
... das "Best Practices" zu verwenden/verbreiten	ja, aber	kann ich mir gut vorstellen	kann ich nicht bewerten	kann ich mir gut vorstellen

Tabelle A.2: Ergebnis Umfrage (Bewertung-Aussagen II)

A.3 Ontologie SVG

Listing A.2: SVG Ontologie Code - Auszug

```
<svg ... > <defs/> <g>
  <path d="M 363 60 L 363 232.13" fill="none" stroke="#000000"
    stroke-miterlimit="10" pointer-events="stroke"/>
  <path d="M 363 237.38 L 359.5 230.38 L 363 232.13 L 366.5 230.38 Z"
    fill="#000000" stroke="#000000"
    stroke-miterlimit="10" pointer-events="all"/>
  <g transform="translate(-0.5 -0.5)">
    <switch>
      <foreignObject style="overflow: visible; text-align: left;"
        pointer-events="none" width="100%">
        <div xmlns="http://www.w3.org/1999/xhtml"
          style="display: flex; align-items: unsafe center;
            justify-content: unsafe center; width: 1px; height:
            1px; padding-top: 149px; margin-left: 363px;">
          <div style="box-sizing: border-box; font-size: 0;
            text-align: center; ">
            <div style="display: inline-block; font-size: 14px;
              font-family: Helvetica; color: #000000;
              line-height: 1.2; pointer-events: all;
              font-style: italic; background-color: #ffffff;
              white-space: nowrap; ">
              {{onto['tag']['topic']['name']}}
            </div>
          </div>
        </div>
      </foreignObject>
      <text x="363" y="153" fill="#000000" font-family="Helvetica"
        font-size="14px" text-anchor="middle"
        font-style="italic">belongs_to
      </text>
    </switch>
  </g>...
</svg>
```

A.4 Wordclouds



Abbildung A.1: Wordclouds Beispieldaten

A.5 Umfrage

Einführung

Umfrage im Rahmen der Bachelor Thesis:

Konzeption und Umsetzung eines Knowledge Graph Prototypen, zur Evaluierung des Einsatzes in der Nachweisdokumentation.

1. Videos angucken:

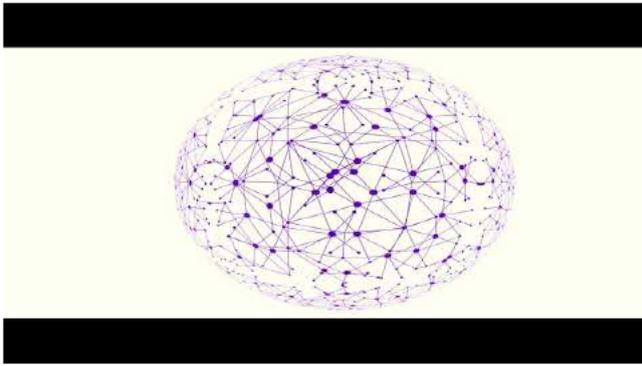
Intro:



<http://youtube.com/watch?v=QdNwGAVydew>

1. Anmerkung?

Woogle:



[http://youtube.com/watch?](http://youtube.com/watch?v=r54kiC6P0uo)

[v=r54kiC6P0uo](http://youtube.com/watch?v=r54kiC6P0uo)

2. Anmerkung?

Praxis

Beim Praxisabschnitt liegt der Fokus auf dem Arbeiten mit der Anwendung. Die Bewertung folgt im Abschnitt 3. <https://woogle.wusel.space/>

3. Wie Bewertest du die Funktionalität des KGs?

Markieren Sie nur ein Oval.

1 2 3 4 5

schwer zu verstehen leicht verständlich

4. Anmerkungen?

Bewertung

Die Kriterien zur Evaluierung sind angelehnt an die Ziele eines Knowledge Management Systems.

5. Der Einsatz eines Knowledge Graphen im Wissensmanagement könnte ...

Wie bewertest du die Aussage?

Markieren Sie nur ein Oval pro Zeile.

	kann ich mir gut vorstellen	ja, aber	hat kein Einfluss	kann ich nicht bewerten
... die Entscheidungen und die Entscheidungsprozesse verbessern	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Reaktionszeiten verringern , zB bei Problemen im Betrieb könnten Verantwortlichkeiten schnell geklärt werden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Projektzeiten verringern, da die Wissensbasis für alle verfügbar wäre und Querverweise nicht verborgen blieben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Prozesse verbessern und könnte neue Arbeitsmethoden bereitstellen!	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Produktivität erhöhen , durch das Entfernen der recherche Barriere.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Produktqualität erhöhen, weil der Fokus auf dem Produkt läge und nicht das 'Nachdokumentieren' .	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Mit dem Einsatz eines Knowledge Graphen im Wissensmanagement ist es möglich ...

Wie bewertest du die Aussage?

Markieren Sie nur ein Oval pro Zeile.

	kann ich mir gut vorstellen	ja, aber	hat kein Einfluss	kann ich nicht bewerten
... den Wissenstransfer zu unterstützen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... neue Wissensgebiete zu entdecken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... das Teamwork zu unterstützen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Ressourcennutzung zu verbessern.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Interne Kommunikation zu unterstützen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... dass Wissen verfügbar zu machen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... die Kreativität zu unterstützen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
... das "Best Practices" zu verwenden/verbreiten"	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

7. Anmerkung:

8. Wie bewertest du den Einsatz eines Knowledge Graph Systems in der Dokumentation (Projekt- / System- / ect-) [1 = hat kein Mehrwert, 5 = ist sehr Sinnvoll]

Markieren Sie nur ein Oval.

	1	2	3	4	5	
hat kein Mehrwert	<input type="radio"/>	ist sehr Sinnvoll				

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.

Google Formulare

A.6 Inhalt der CD-ROM

Dieser Arbeit ist eine CD-ROM beigelegt. Unter dem Wurzelverzeichnis der CD-ROM befinden sich Ordner und Dateien, siehe Listing.

Listing A.3: README Markdown Datei mit dem Inhalt der CD-ROM

Bachelor These - Inhalt CD-Rom

Dateien:

Bachelor-These.pdf (Bachelor Arbeit)
docker-compose.yml (startet KG-Backend + KG-Frontend + neo4j-Datenbank)
README.md

Ordner:

- * kg-backend/ - Knowledge Graph Engine
- * knowledge-graph-client/ - Knowledge Graph UI
- * data/ - Beispieldaten

Software - Setup

Voraussetzungen lokale Umgebung:

- * docker
- * docker-compose

Um die Knowledge Graph Software zu starten kann der folgende Befehl im Terminal eingegeben werden:

```
$ docker-compose up
```

Nach dem Start sind die Komponenten unter den folgenden Adressen zu erreichen:

```
$ http://localhost:3000 - Backend (mit UI der einzelnen module zum testen)
  Password="hello"
$ http://localhost:3010 - Fronten (woogle) Password="hello"
$ http://localhost:7474 - Datenbank UI Password="hell"
```

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:

Konzeption und Umsetzung eines Knowledge Graphen zur Evaluierung des Einsatzes in der Nachweisdokumentation

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