HTW BERLIN

WIRTSCHAFTSWISSENSCHAFTEN II

Lehrstuhl für Wirtschaftsinformatik

**Wissensmodellierung und semantische Suche am Beispiel einer Projektdatenbank für IoT Projektmanager**

Version 0.5

Betreuer: Prof. Dr. rer. nat. Holger Hemling

Dr. Frank Puhlmann

Abgegeben von: Christiane Gießler (s0531050)

Studiengang: Wirtschaftsinformatik (Master)

Abgabetermin: 10.09.2015

Inhaltsverzeichnis

[1. Einleitung 1](#_Toc427140870)

[1.1 Problemstellung 1](#_Toc427140871)

[1.2 Vorgehen 1](#_Toc427140872)

[2. Begrifflichkeiten 2](#_Toc427140873)

[2.1 Wissen 2](#_Toc427140874)

[2.1.1 Definition 2](#_Toc427140875)

[2.1.2 Wissensmanagement 4](#_Toc427140876)

[2.1.2.1 Komponenten 4](#_Toc427140877)

[2.1.2.2 Kernprozesse 5](#_Toc427140878)

[2.1.2.3 Anwendungsbereiche 7](#_Toc427140879)

[2.1.3 Wissensorganisationssysteme 7](#_Toc427140880)

[2.1.3.1 Arten von Wissensorganisationsystemen 8](#_Toc427140881)

[2.1.4 Wissensmodellierung 12](#_Toc427140882)

[2.2 Datenbanken 15](#_Toc427140883)

[2.2.1 Definition 15](#_Toc427140884)

[2.2.2 Datenbankmodelle 16](#_Toc427140885)

[2.2.2.1 Relationale Datenbankmodelle 16](#_Toc427140886)

[2.2.2.2 Dokumentenorientierte Datenbankmodelle 16](#_Toc427140887)

[2.2.2.3 Graph Datenbanken 16](#_Toc427140888)

[3. Wissen in Zeiten des Internet der Dinge 17](#_Toc427140889)

[3.1 Die Entwicklung des Web zum Internet der Dinge 17](#_Toc427140890)

[3.1.1 Evolution des Web zum Web 3.0 17](#_Toc427140891)

[3.2 Wissensorganisationssysteme am Beispiel von Tagging 19](#_Toc427140892)

[3.2.1 Definition 19](#_Toc427140893)

[3.2.2 Automatisiertes Tagging 20](#_Toc427140894)

[3.2.3 Manuelles Tagging 21](#_Toc427140895)

[3.2.3.1 Tagging durch Experten 21](#_Toc427140896)

[3.2.3.2 Tagging durch den Endnutzer 21](#_Toc427140897)

[3.2.4 Social Tagging / Folksonomy 21](#_Toc427140898)

[3.2.5 Gegenüberstellung automatisiertes und manuelles Tagging 22](#_Toc427140899)

[3.2.6 Anwendungsbereiche 23](#_Toc427140900)

[3.3 Wissenssuche mithilfe semantischer Werkzeuge 24](#_Toc427140901)

[3.3.1 (Web 3.0) 24](#_Toc427140902)

[3.3.2 Semantic Web 24](#_Toc427140903)

[3.3.3 Semantische Suche 26](#_Toc427140904)

[3.3.3.1 (Semantische Technologien) Technologische Verfahren zur Suchanfragenverarbeitung 26](#_Toc427140905)

[3.3.3.2 Optimierungsmöglichkeiten einer Suchanfrage 26](#_Toc427140906)

[3.4 Das Internet der Dinge als Resultat des Web 3.0 (evtl. muss das hoch) 28](#_Toc427140907)

[3.4.1 Definition & Aktualität des Begriffs 28](#_Toc427140908)

[3.4.2 Ignite Projektmethodik 29](#_Toc427140909)

[4. Entwicklung eines Wissensorganisationssystems 31](#_Toc427140910)

[4.1 Modellierung des Wissensorganisationssystems 31](#_Toc427140911)

[4.1.1 Eingrenzung von Rahmen, Zweck und Umfang 31](#_Toc427140912)

[4.1.2 Wortfindungen, Bilden von Relationen und Klassifizierungen 35](#_Toc427140913)

[4.1.3 Bilden von Relationen 37](#_Toc427140914)

[4.1.4 Bilden von Klassifizierungen 38](#_Toc427140915)

[4.1.5 Vervollständigung 39](#_Toc427140916)

[4.2 Mögliche Umsetzungsszenarien 39](#_Toc427140917)

[4.2.1 Marktanalyse 39](#_Toc427140918)

[4.2.1.1 Kriterien der Marktanalyse 39](#_Toc427140919)

[4.2.1.2 Ergebnis der Marktanalyse 40](#_Toc427140920)

[4.2.1.3 Detaillierte Betrachtung ausgewählter Software 40](#_Toc427140921)

[4.2.2 Gegenüberstellung verschiedener Datenbankmodelle 42](#_Toc427140922)

[4.2.2.1 Relationale Datenbank am Beispiel von mySQL 42](#_Toc427140923)

[4.2.2.2 Graph Datenbank am Beispiel von Neo4j 44](#_Toc427140924)

[4.2.3 Ergebnis 45](#_Toc427140925)

[4.3 Entwurfsphase des Wissensorganisationssystems 45](#_Toc427140926)

[4.3.1 Domänenmodell der Projektdatenbank 45](#_Toc427140927)

[4.3.2 Mock-Ups der Anwendung 47](#_Toc427140928)

[5. Prototypische Umsetzung 50](#_Toc427140929)

[5.1 Softwarearchitektur 51](#_Toc427140930)

[5.2 Entwicklung des Backends 51](#_Toc427140931)

[5.2.1 Anlegen der Datenbank & Grunddatensatz implementieren 51](#_Toc427140932)

[5.2.2 Datenimplementierung 52](#_Toc427140933)

[5.2.2.1 Phase 1: Datenimplementierung mithilfe von „Learning Neo4j” 52](#_Toc427140934)

[5.2.2.2 Phase 2 55](#_Toc427140935)

[5.2.3 Abfragemöglichkeiten mit Cypher 56](#_Toc427140936)

[5.3 Entwicklung des Frontends 59](#_Toc427140937)

[5.3.1 Verwendete Komponenten (Angular JS) 59](#_Toc427140938)

[5.3.1.1 AngularJS 60](#_Toc427140939)

[6. Evaluation 61](#_Toc427140940)

[6.1 Vorgehen 61](#_Toc427140941)

[6.2 Durchführung 61](#_Toc427140942)

[6.3 Auswertung 61](#_Toc427140943)

[7. Zusammenfassung 62](#_Toc427140944)

[7.1 Fazit 62](#_Toc427140945)

[7.2 Kritische Betrachtung 62](#_Toc427140946)

[7.3 Nächste Schritte 62](#_Toc427140947)

[8. Referenzen 63](#_Toc427140948)

[9. Begrifflichkeiten 64](#_Toc427140949)

[Anhang 65](#_Toc427140950)

[9.1 Anhang A 65](#_Toc427140951)

# Einleitung

Projektdatenbank: Das Prinzip zum Referenzieren auf andere Projekte wurde bereits in einfacher Form in einer Excel-Tabelle umgesetzt. Dabei sind kategorisierte Aussagen über verschiedene IoT-Projekte vorhanden. Diese ergeben, je nach Angaben, Spinnendiagramme (wie in den Projektdimensionen ersichtlich).

## Problemstellung

Im privaten Umfeld wurden Tagging-Systeme in Form von Folksonomies bereits vollständig integriert und finden ihre Anwendung im Alltag. Die Kombination aus Nutzern, die Tags (Schlagworte) an Ressourcen hinzufügen, und deren immense Anzahl führen zu einer nicht bestimmbaren Anzahl an (auswertbaren) Daten. Dies bildet ein, wenn nicht das Erfolgskonzept von Twitter, Flickr und Co., die sich durch Hashtags an Bildern und Texten auszeichnen.

Hierbei stellt sich die These, ob sich das Prinzip der Tagging-Systeme nicht nur für den Privatanwender nutzen lässt oder ob es in Zeiten einer immer enger vernetzten Welt nicht auch im Wirtschaftsbereich genutzt werden kann. Können speziell im noch jungen Bereich des Internet der Dinge Folksonomies hilfreich sein, um neues Wissen schneller/ einfacher/ besser/ eher zu verstehen? Was bringt die Einbindung des Nutzers in solch ein System?

Welche Datenbank ist für das Tagging-System sinnvoll?

## Vorgehen

# Begrifflichkeiten

Nachfolgend werden die Grundbegriffe des Wissens sowie Datenbanken näher betrachtet.

## Wissen

…

### Definition

Die Definition des Begriffs „Wissen“ wird in verschiedensten wissenschaftlichen Bereichen benötigt und dementsprechend interpretiert. So ist es beispielsweise entscheidend, ob die Definition aus einem soziologischen Umfeld kommt oder aus der Informationstechnologie.

Da sich diese Thesis mit der Informationstechnologie beschäftigt, ist die nachfolgende Definition die geeignetste:

„Wissen bezeichnet die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen […]“. Dabei darf nicht außer Acht gelassen werden, dass Wissen nicht allein zur Problemlösung notwendig ist, sondern dadurch automatisch auch Entscheidungen unterstützt und getroffen werden können. Weiterhin heißt es: „[…] Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden. Es wird von Individuen konstruiert und repräsentiert deren Erwartungen über Ursache-Wirkungs-Zusammenhänge“. Wie Anfangs erwähnt ist die Definition, also auch das Wissen über den Begriff „Wissen“ demnach abhängig vom Individuum, welches es erfasst.

Nicht benannt ist in dieser Definition, dass zum Wissen mehr gehört als Kenntnisse, Fähigkeiten und Erwartungen. Hinzu kommen Fakten, Konzepte, Theorien, Heuristische Methoden, Prozeduren und Beziehungen [Bodrow].

Doch wie entsteht Wissen? Es ist nicht einfach vorhanden, sondern wird erst generiert. Visuell kann dies aus der Wissenstreppe von NORTH abgeleitet werden:

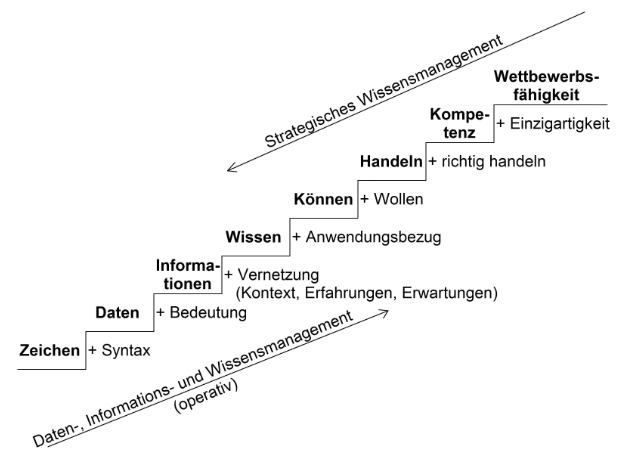


Abbildung Die Wissenstreppe nach North1999 (S.37 ???)

Nach North kann Wissen in Form einer Treppe abgegrenzt werden.

Zu Beginn dieser Treppe stehen Zeichen, zum Beispiel alphanumerische. Eine Folge dieser Zeichen, die den Formalien eines Regelsatzes (also einer Syntax) entsprechen, ergeben Daten. Ein Beispiel hierbei sind die Zeichenfolgen einer ISBN-Nummer. Bedingungen in Form der Syntax sind dabei, dass die ISBN-Nummer aus zehn oder 13 Zeichen in Form von Zahlen besteht. Um Daten als Information zu übermitteln, wird diesen zusätzlich eine Bedeutung übergeben. Zum Beispiel die Information: „Das vorliegende Buch wurde unter dem Namen ‚Enterprise IoT‘ im Jahr 2014 erstmals unter anderem vom Dr. Frank Puhlmann veröffentlicht. Die ISBN-Nummer des Buches ist die nachfolgende: […]“. Um diese Information zuletzt in Wissen umzuwandeln, ist eine zusätzliche Vernetzung dessen nötig, in dem ein Kontext hinzugefügt, Erfahrungen und Erwartungen benannt werden. Zum Beispiel: „Aufgrund der Erkenntnis, dass […] wurden mehrere Case-Studies mit IoT-Projektleitern aus aller Welt durchgeführt. Aus den gewonnenen Informationen konnten im Gesamten betrachtet neue Erkenntnisse in Form von Wissen über den noch neuen Bereich des Internet der Dinge gewonnen werden. So zum Beispiel, dass […]“.

Die Begriffe „Können“, „Handeln“; „Kompetenz“ und „Wettbewerbsfähigkeit“ beziehen sich innerhalb der Wissenstreppe auf den Bereich des strategischen Wissensmanagements und werden im Kontext der Master Thesis nicht genauer betrachtet.

### Wissensmanagement

Um den Begriff des Wissensmanagements zu verstehen, ist eine vorherige Betrachtung der Komponenten von Wissen nötig (wirklich?). Desweiteren werden Kernprozesse und Anwendungsbereiche genauer vorgestellt.

#### Komponenten

Die Komponenten von Wissensmanagement bestehen aus den Teilkomponenten Mensch, Organisation und Technik (siehe Abbildung 2)

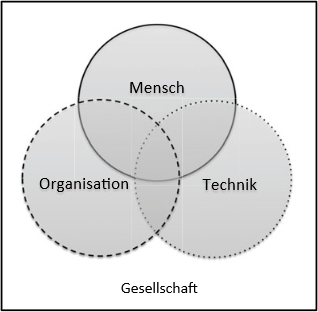


Abbildung Komponenten von Wissen nach http://link.springer.com/book/10.1007%2F978-3-658-08195-9

…

|  |
| --- |
| Die erste Komponente „Mensch“  bezieht sich auf die einzelnen Organisationsmitglieder, die Wissensträger, und deren Kenntnisse, Fähigkeiten und Kompetenzen sowie deren Motivation und Einstellungen  (Reinmann-Rothmeier & Mandl, 2000). Die Träger relevanten Wissens  sind die eigentlichen „Triebfedern“ kontinuierlicher Lernprozesse und  bilden den Kern jedes Wissensmanagements (Reinmann-Rothmeier et al.,  2001, S. 18). Auch in einer europaweiten Unternehmensbefragung konnte  festgestellt werden, dass für den Wissensmanagementerfolg individuelle  Faktoren wie die Fähigkeiten und Motivation der Mitarbeiter wichtig  sind (Heisig, 1999).  Die zweite Komponente „Organisation“ ist für die Strukturen, Prozes-  se und Rahmenbedingungen verantwortlich. Sie soll die Entwicklung  einer wissens- und lernförderlichen Umgebung sowie Kultur im Unter-  nehmen beeinflussen, um den Umgang mit der Ressource Wissen zu  erleichtern (Reinmann-Rothmeier & Mandl, 2000; Mandl, 2010). Bezogen  auf die Ergebnisse der Unternehmensbefragung wurden ebenfalls orga-  nisationale Aspekte, wie die Organisationskultur und Unterstützun  durch das Management, für den Erfolg von Wissensmanagement artiku-  liert (Heisig, 1999).  Die dritte Komponente „Technik“ befasst sich mit geeigneten Informations- und Kommunikationsinfrastrukturen sowie effizienten und  nutzerfreundlichen technischen Werkzeugen (Reinmann-Rothmeier &  Mandl, 2000; Mandl, 2010). Die Informationstechnologie wurde ebenfalls  als wichtiges Gestaltungsfeld des Wissensmanagements identifiziert  (Heisig, 1999). |
| http://link.springer.com/book/10.1007%2F978-3-658-08195-9 |

#### Kernprozesse

Wissensmanagement besteht aus den nachfolgenden Kernprozessen:

* Wissensidentifikation
  + Die Identifikation von Wissen nutzen Unternehmen, um ihr Wissen bezüglich interner und externer Daten, sowie Informationen und Fähigkeiten zu erfassen (vgl. Probst S. 29)
* Wissensbewahrung
* Wissensnutzung
* Wissensverteilung
* Wissensentwicklung
  + Die Entwicklung von Wissen beschreibt den „ allgemeine[n] Umgang des Unternehmens mit neuen Ideen und die Nutzung der Kreativität der Mitarbeiter“ (Probst S.29).
* Wissenserwerb
  + Der Erwerb von Wissen ist geprägt durch externe Wissensträger sowie der Nutzung neuer Wissensquellen.



Abbildung Kernprozesse des Wissensmanagements (Probst, S. 30 ff)

In Abbildung 3 sind alle Kernprozesse des Wissensmanagements und deren wechselseitige Beziehung zueinander dargestellt. Durch diese Darstellung wird verdeutlicht, dass sich alle Kernprozesse separat ändern können, dies aber auch jeweils Auswirkung auf alle anderen Prozesse hat.

Die Kernprozesse des Wissensmanagements stehen sich wechselseitig gegenüber.

|  |
| --- |
| Jeder Kernprozess kann einerseits separiert verändert sowie verbessert werden, andererseits wirkt sich eine Modifikation immer auch auf die anderen Kernprozesse aus und sollte deshalb vermieden werden (vgl. Probst et al. 2010:28). Der Prozess Wissensidentifikation ermöglicht einem Unternehmen „den überblick über interne und externe Daten, Informationen und Fähigkeiten zu behalten“ (Probst et al. 2010:29). „Die Schaffung von Wissenstransparenz verdeutlicht bestehende Wissenslücken und schafft Voraussetzungen, um über Wissenserwerb und Wissensentwicklung zu entscheiden.“ (Probst et al. 2010:89) Der Prozess des Wissenserwerbs zielt auf die Rekrutierung von externen Wissensträgern ab und beschäftigt sich mit dem Kultivieren neuer Wissensquellen (vgl. Probst et al. 2010:29). Neu akquiriertes Wissen sollte dabei zu bereits verfügbarem Wissen passen, um einer möglichen Ablehnung durch die Mitarbeiter vorzubeugen (vgl. Probst et al. 2010:109). Der Prozess der Wissensentwicklung definiert den „[…] allgemeine[n] Umgang des Unternehmens mit neuen Ideen und die Nutzung der Kreativität der Mitarbeiter […] (Probst et al. 2010:29). „Interaktion, Kommunikation sowie Transparenz und Integration bilden die Schlüsselgrößen der kollektiven Wissensentstehung.“ (Probst et al. 2010:138) |
| Bei Probst schauen und richtig zitieren |

#### Anwendungsbereiche

(Aus Buch Wissensmanagement-Unterstützung für die Entwicklung und Nutzung von Prozessmodellen als wissensvermittelnde Artefakte)

Expertenverzeichnisse und Skill-Management

Persönliches und gruppenbezogenes Informationsmanagement

Wissensmanagement und Web 2.0 (!!)

Expertensysteme

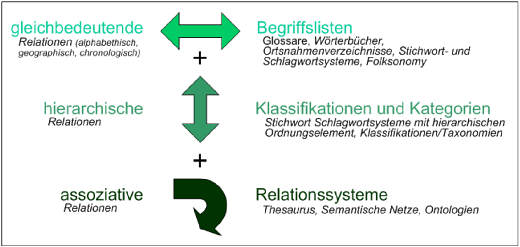
DEFINITION WISSENSMANAGEMENTSYSTEME!!!

### Wissensorganisationssysteme

Wissensmanagementsysteme oder auch Wissensorganisationssysteme (kurz: WOS) sind Systeme, welche Informationen in gewisser Weise gliedern, um so Wissen zu generieren und zu speichern. Sie haben zum Ziel, Wissen zu Ordnen, Informationen wiederzufinden und innerhalb des Systems eine nachvollziehbare Navigation abbilden zu können (vgl. Social Semantic Web S. 128). Dabei gibt es vor allem zwei Typen, die zu beachten sind: Ordnungstypen und Funktionstypen.

Ordnungstypen zeigen verschiedene Optionen auf, wie eine Beziehung dargestellt werden kann. Dementsprechend werden WOS danach kategorisiert. Bestimmte Ordnungstypen sind: das Alphabet (z.B. Wörterbücher), Kategorien (z.B. Enzyklopädien), geographische Orte (z.B. Atlanten), die Zeit (z.B. bei Anleitungen) sowie eine Hierarchie (z.B. die Ordnung von klein zu groß).

Relationstypen zeigen Verhältnisse einzelner Objekte auf. Dabei können die nachfolgenden unterschieden werden: gleichbedeutende Relationen (z.B.: Begriffslisten), hierarchische Relationen und assoziative Relationen. Gleichbedeutende und hierarchische Relationen ergeben in Kombination Klassifikationen und Kategorien. Kommen assoziative Relationen hinzu entstehen Relationssysteme. (vgl. SSW S. 129f).



#### Arten von Wissensorganisationsystemen

🡪 Stichwort: (Semantische) Darstellung von Wissen

##### Taxonomie

Eine Taxonomie beschreibt die hierarchische Untergliederung von Begriffen (vgl. Whitepaper Wissensgarten S.2). Im informationstechnischen Raum ist die Ordnerstruktur eines Betriebssystems ein typisches Beispiel. Dabei können Begriffe immer Unterbegriffe haben. Diese Unterbegriffe sind genau einem Oberbegriff zugeordnet. Am Beispiel von Wissensgarten in Abbildung 3 wird deutlich, dass die Begriffe F-Logic und Ontologie zum Oberbegriff Semantik gehören. Allerdings wird nicht deutlich, dass der Begriff „PhD Student“ ein Synonym für „Doktorrand“ darstellt.

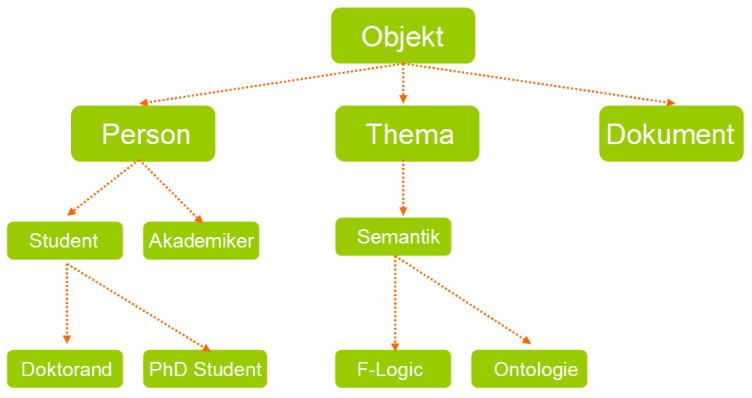


Abbildung Beispiel einer Taxonomie nach Wissensgarten

##### Thesaurus

Ein Thesaurus „stellt eine Terminologie zu einer bestimmten Wissensdomäne dar“ (siehe Wissensgarten S.3). Dabei wird wie in der Taxonomie eine Hierarchie angewandt. Hinzukommen die Möglichkeiten, Synonyme und Ähnlichkeiten aufzuzeigen. Wie in ersichtlich ist es in diesem Beispiel möglich, „PhD Student“ als Synonym für Doktorrand zu bezeichnen und F-Logic als Ähnlichkeit zur Ontologie hervorzuheben.

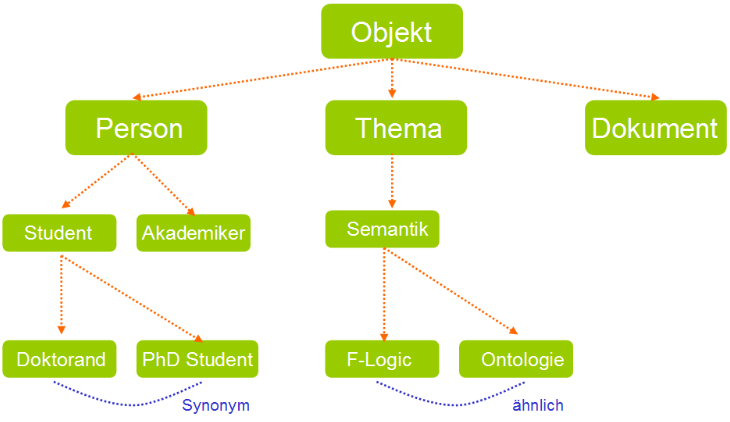


Abbildung Beispiel eines Thesaurus nach Wissensgarten

##### Topic Map

Die Topic Map wurde als internationaler Industriestandard in der IS0 13250 definiert. Zudem wurde sie später als XTM (XML Topic Maps) in XML beschrieben. Sie beinhaltet „Subjects (abstrakte Dinge), Topics (deren addressierbare Repräsentationen), Assoziationen, Scopes (Gültigkeitsbereiche für Topics) und zugeordnet[e] Dokument[e] außerhalb der Topic Maps (Occurences)“(Siehe Wissensgarten S.4).

In Abbildung 5 wird die Erweiterung im Gegensatz zum Thesaurus deutlich: Neben der Hierarchie und der Möglichkeit, Synonyme und Ähnlichkeiten zu verdeutlichen, können in der Topic Map zusätzlich Dokumente angehangen werden (externe und Websiten). Ebenso ist es möglich, dass nun zwischen allen Begriffen assoziiert werden kann.

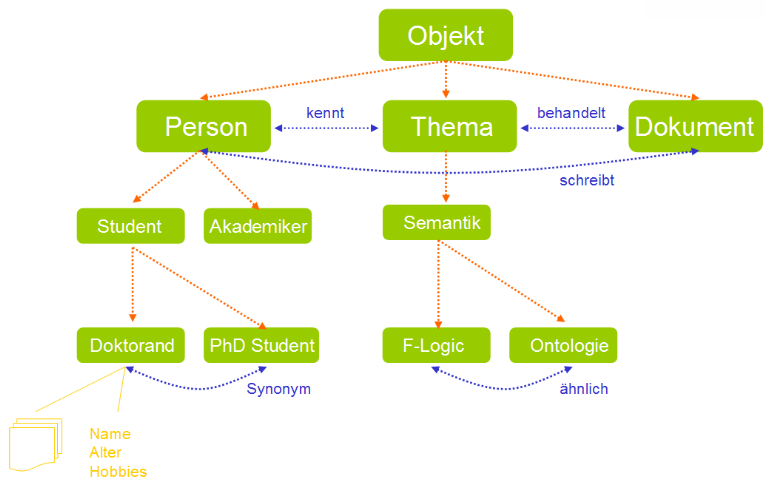


Abbildung Beispiel einer Topic Map nach Wissensgarten

🡪 http://www.ontopia.net/index.jsp

##### Ontologie

„Der große Unterschied zur Taxonomie ist der, dass die Ontologie ein Netzwerk von Informationen mit logischen Relationen darstellt, während die Taxonomie eine hierarchische Untergliederung bildet.“ Anders ausgedrückt bildet die Ontologie ein semantisches Netz, wohingegen die Taxonomie eine reine Baumansicht darstellt.

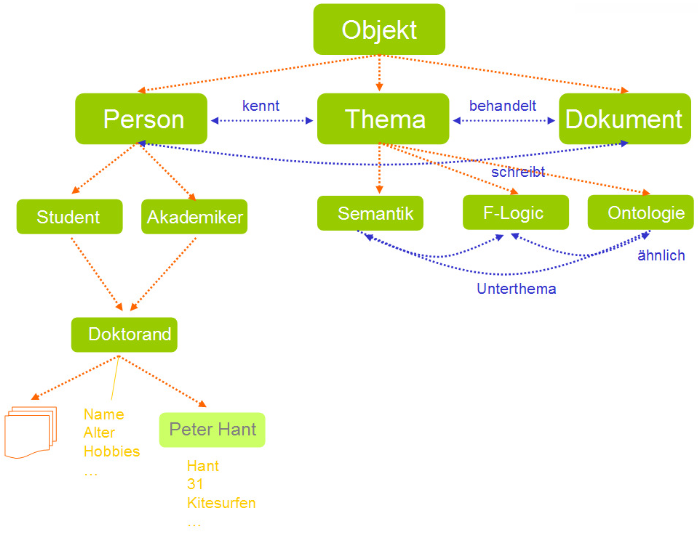


Abbildung Beispiel einer Ontologie nach Wissensgarten

Bei der Ontologie wird im Gegensatz zur Topic Map zwischen einer „is-a“ und einer „has-a“-Hierarchie unterschieden. „Is-a“ steht für eine Vererbungshierarchie. „Has-a“ greift dafür die Relationen auf. Schema und Inhalte können dadurch voneinander getrennt werden.

„Die Konzepte einer Ontologie definieren das Schema, während Instanzen, die jeweiligen Ausprägungen darstellen“.

„Die Ontologie verfügt über ein mächtiges Regelwerk, mit dem Zusammenhange zwischen Objekten der Ontologie (und auch anderer Ontologien) mittels „Wenn-Dann“ Beziehungen, Zuweisungen, Negationen, logischen Verknüpfungen, mathematischen Operationen und weiteren Funktionen ausgedrückt werden können“ (Wissensgarten S.5).

##### Zusammenfassung

### Wissensmodellierung

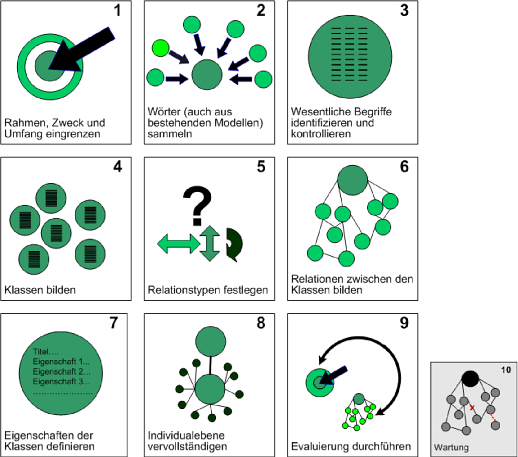
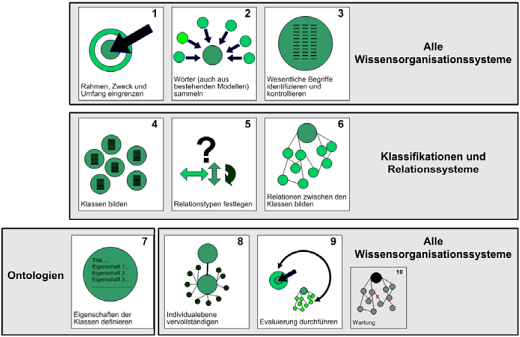


Abbildung Phasen der Wissensmodellierung nach Buch Social Semantic Web

//

 (s. 140)

Im Buch Social Semantic Web wird ein generischer Ansatz vorgestellt, Wissen nach zehn Phasen zu modellieren, wobei die Phasen 4 bis 7 je nach System eingesetzt werden. Ein Überblick über die Wissensmodellierung und die Untergliederung in einzelne Phasen ist Abbildung 7 zu entnehmen.

/ Nachfolgend wird die Generierung in Form von Phasen zur Wissensmodellierung nach dem Buch (+Seiten) … vorgestellt.

*Phase 1: Rahmen, Zweck und Umfang eingrenzen*

Zunächst müssen der Rahmen, der Zweck und der Umfang des Wissensorganisationssystems eingegrenzt werden. Es werden dabei Fragen zu den Unterpunkten Kontext, Benutzer, Inhalt und Vokabular gestellt.

*Phase 2: Wörter (auch aus bestehenden Modellen) sammeln*

Bei allen WOS, die erstellt werden, können bestimmte Wörter wiederverwendet werden. „Abhängig von gewünschten Umfang, Sprachstil und Detaillierung kann auf folgende Quellen zurückgegriffen werden:

Metadaten von Dokumenten

Struktur (Navigationsleiste, Sitemap) von Websiten

Bestehende Wissensorganisationssysteme […]

Aktuelle Fachliteratur (und die darin vergebenen Schlagwörter)

[…]

Befragung von potentiellen Benutzern und Experten

Aber auch bestehende Datenbanken und Tabellen.“ (SSM S. 141)

*Phase 3: Wesentliche Begriffe identifizieren und kontrollieren*

Alle in Phase 2 gesammelten Wörter werden in Phase 3 in ein einheitliches Schema übernommen. Dafür werden die Begriffe einheitlich in Singular oder Plural geschrieben, Abkürzungen und Fremdwörter werden aufgelöst, Dopplungen aufgelöst sowie Synonyme und Homonyme angeglichen. Ebenso hilft es, fokussierte Fragen zu definieren, die das spätere WOS beantworten soll.

Entweder wird nun eine nach dem Alphabet sortierte Begriffsliste erstellt. Oder es werden (bei Verwendung von hierarchischen oder assoziativen Relationen) Klassen gebildet (vgl. SSW S. 142). Tritt dieser Fall ein, müssen die Phasen 4 bis 7 zusätzlich bearbeitet werden. Wird eine Begriffsliste erstellt, können die Phasen übersprungen und bei Phase 8 wieder eingestiegen werden.

*Phase 4: Klassen bilden*

Die in Phase 3 selektierten Begriffe werden nun strukturiert. Dabei werden Klassen identifiziert und in eine hierarchische Struktur gebracht. Dafür ist Phase 5 notwendig.

*Phase 5: Relationstypen festlegen*

Je nach Relationstyp ändert sich die Art des WOS. Möglich sind:

„Klassifikationen: Hierarchische Beziehungen (Oberbegriff, Unterbegriff)

Thesaurus: BF (Benutzt für), BS (Benutze Synonym), OB (Oberbegriff), UB (Unterbegriff) und VB (Verwandter Begriff)“ (SSW S. 143)

Semantische Netze und Ontologien mit beliebigen Relationen

*Phase 6: Relationen zwischen den Klassen Bilden*

Je nach ausgewählten Relationstyp werden die Klassen aus Phase 4 nun miteinander verbunden.

*Phase 7: Eigenschaften der Klassen definieren*

Jeder Klasse müssen Relationen und Attribute zugeordnet werden. Diese können beispielsweise nach äußeren und inneren Attributen sowie nach Beziehungen zu anderen Klassen geordnet werden.

*Phase 8: Individualebene vervollständigen*

Bis zur Phase 7 wurde ein Schema für das WMO erstellt. Um daraus eine Wissensbasis zu generieren, müssen Dokumente, Objekte und Ausprägungen den benannten Begriffen zugeordnet werden (vlg. SSW 144).

*Phase 9: Evaluierung durchführen*

In Phase 9 werden vorhandene WOS nach bestimmten Kriterien evaluiert. Dafür können beispielsweise die genannten Fokusfragen aus Phase 3 genutzt werden. Ebenso ist es möglich zukünftige Nutzer ausgewählte WOS nach gewissen Mustern zu testen.

*Phase 10: Wartung*

Nach der Evaluierung des WOS und der Implementierung der Daten kann das System eingesetzt werden. Eine regelmäßige (manuelle) Wartung ist dabei notwendig, um das WOS auf einem aktuellen Stand zu halten.

## Datenbanken

Nachfolgend werden der Begriff der „Datenbanken“ definiert und verschiedene Datenbankmodelle vorgestellt.

### Definition

|  |  |
| --- | --- |
| <http://www.itwissen.info/definition/lexikon/Datenbank-DB-database.html> >  Eine Datenbank (DB) kann als organisierte Datensammlung verstanden werden, die einem zentralen Zweck dient. Die Organisation der [Daten](http://www.itwissen.info/definition/lexikon/Daten-data.html) umfasst deren Speicherung, [Formatierung](http://www.itwissen.info/definition/lexikon/formatting-Formatierung.html), Verwaltung, Zugriff, Manipulation und Darstellung in konsistenter Form.  Eine Datenbank ist ein elektronisches [Archiv](http://www.itwissen.info/definition/lexikon/Archivierung-archiving.html), das sich durch einen raschen Zugriff und eine Platz sparende Aufbewahrung großer Datenmengen auszeichnet. Daten oder [Nachrichten](http://www.itwissen.info/definition/lexikon/message-Nachricht-MSG.html) können nach variablem Ordnungsschema einzeln oder verknüpft sowohl für den [Dialogbetrieb](http://www.itwissen.info/definition/lexikon/Dialogbetrieb-conversational-mode.html) als auch für die [Stapelverarbeitung](http://www.itwissen.info/definition/lexikon/Stapelverarbeitung-BOS-batch-operating-system.html) ausgelegt sein bzw. abgerufen werden (nach [DIN 44300](http://www.itwissen.info/definition/lexikon/DIN-44-300.html) für die Verwaltung der enthaltenen Daten bzw. [Dateien](http://www.itwissen.info/definition/lexikon/Datei-file.html)). Eine Datenbank kann von mehreren Anwendern und Anwendungsprogrammen gemeinsam genutzt werden. Man unterscheidet die Anordnung der Daten unter Berücksichtigung der optimalen Speichernutzung und die logische Datenstruktur, die sich aus der Beziehung zwischen zusammengehörigen Daten ergibt. Dadurch werden Daten unabhängig von den Anforderungen an ihre jeweilige Anwendung nach einheitlichen Gesichtspunkten verwaltet.  Strukturen von Datenbanken.  Neben den textbasierten Datenbanken gibt es solche die neben [Text](http://www.itwissen.info/definition/lexikon/Text-text.html) und [Numerik](http://www.itwissen.info/definition/lexikon/Numerik-numerical.html) auch [Grafiken](http://www.itwissen.info/definition/lexikon/Grafik-graphics.html), Bilder, [Audio](http://www.itwissen.info/definition/lexikon/Audio-audio.html), [Video](http://www.itwissen.info/definition/lexikon/Video-video.html) und zusammengesetzte [Dokumente](http://www.itwissen.info/definition/lexikon/Dokument-DOC-document.html) verwalten können. Je nach Strukturierung und Anwendung der Datenbank unterscheidet man zwischen [verteilten Datenbanken](http://www.itwissen.info/definition/lexikon/Verteilte-Datenbank-DDB-distributed-database.html), [relationalen Datenbanken](http://www.itwissen.info/definition/lexikon/relational-database-management-system-RDBMS-Relationales-Datenbank-Managementsystem.html), öffentlichen, objektorientierten und [objektrelationalen Datenbanken](http://www.itwissen.info/definition/lexikon/Objektrelationale-Datenbank-object-relational-database.html).  Die [Abfrage](http://www.itwissen.info/definition/lexikon/Abfrage-query.html) von Datenbanken erfolgt mit [Abfragesprachen](http://www.itwissen.info/definition/lexikon/query-language-QL-Abfragesprache.html), von denen die bekannteste für relationale Datenbanken die [Structured Query Language](http://www.itwissen.info/definition/lexikon/structured-query-language-SQL.html) (SQL) ist. Eine [Implementierung](http://www.itwissen.info/definition/lexikon/Implementierung-implementation.html) von SQL findet in [MySQL](http://www.itwissen.info/definition/lexikon/MySQL.html) statt. Eine andere ist [Multidimensional Expression](http://www.itwissen.info/definition/lexikon/MDX-multidimensional-expression-Multidimensionaler-Ausdruck.html) (MDX), die in multidimensional aufbereiteten Datenmengen als Suchwerkzeug eingesetzt wird.  Was die Datenbank-Programmierung betrifft, so gibt es für alle modernen [Programmiersprachen](http://www.itwissen.info/definition/lexikon/Programmiersprache-PL-programming-language.html), wie [C++](http://www.itwissen.info/definition/lexikon/C-plus-plus.html), [Java](http://www.itwissen.info/definition/lexikon/Java.html), [Hypertext Preprocessor](http://www.itwissen.info/definition/lexikon/hypertext-preprocessor-PHP.html) (PHP) und [PEARL](http://www.itwissen.info/definition/lexikon/PEARL-process-and-experiment-automation-realtime-language.html) [Bibliotheken](http://www.itwissen.info/definition/lexikon/Bibliothek-library.html) welche die Datenbank-Programmierung ermöglichen.   |  | | --- | | <http://link.springer.com/chapter/10.1007/978-3-322-96183-9_7> \*von 89  Eine Datenbank ist eine strukturierte Ansammlung von Informationen. Beispiele sind ein einfaches Telefonbuch oder Datenbanken für Lageroder Adreßverwaltung. | |

### Datenbankmodelle

Datenbankmodelle können hierarchisch, netzwerkartig, relational, objektrelational, objektorientiert und/ oder dokumentenorientiert aufgebaut sein. Um das Themengebiet einzugrenzen, werden nachfolgend die relationalen und dokumentenorientierten Modelle beispielhaft vorgestellt, da diese bei der späteren Evaluierung Berücksichtigung finden und die Gegensätze zwischen Datenbankmodellen mit und ohne Schema darstellen.

#### Relationale Datenbankmodelle

#### Dokumentenorientierte Datenbankmodelle

#### Graph Datenbanken

[**ZUSATZINFORMATION**](file:///C:\Users\gic1be\Dropbox\Master%20Thesis\Geschriebenes\Zusatzinformation%20Graph%20Datenbanken.docx)

## Projektmanagement

### Definition

#### Projekt

#### Management

#### Projektmanagement

### …

### Projektmanagement und Wissen (in Kontext MT)

### Beispielanwendung von Wissen = PM

“Projektmanagement ist die Anwendung von Wissen, Fertigkeiten, Werkzeugen und Methoden auf Projektvorgänge, um die Projektanforderungen zu erfüllen.“ - Project Management Institute (PMI)

# Wissen in Zeiten des Internet der Dinge

Nachdem Grundbegriffe des Wissens und der Datenbanken beschrieben wurden, geht das nachfolgende Kapitel genauer auf Veränderungen in der heutigen Zeit ein. Dabei wird die Entwicklung des Web zum Internet der Dinge untersucht. Anschließend wird die semantische Wissenssuche betrachtet. Zuletzt wird das Internet der Dinge als Resultat des Web 3.0 vorgestellt.

## Die Entwicklung des Web zum Internet der Dinge

Aktuell werden Buzzwords wie Industrie 4.0 verwendet…

Nachfolgend wird aufgezeigt, wie der aktuelle Stand der Technik im Sinne von Web 3.0 ist und wie dessen vorherige Evolution entstanden ist. Daraufhin wird auf die einhergehende Semantik eingegangen, mithilfe deren Werkzeuge erweiterte Suchtechnologien vorgestellt werden. Im Anschluss wird das Internet der Dinge und deren einhergehende Herausforderungen an die heutige Wirtschaft beschrieben. Daraufhin werden Wissensmodellierungssystem im Web 3.0 analysiert.

### Evolution des Web zum Web 3.0

Das Web hat seinen Ursprung im Jahr 1989 als Projekt am CERN-Institut in der Schweiz, bei dem unter anderem Tim Berners-Lee HTTP und HTML entwickelte. Sein Gedanke an das World Wide Web war folgender:

„The World Wide Web (W3) is a wide-area hypermedie information retrieval initiative aming to give universal acces to a large universe of documents.“ (Quelle suchen)

Ziel war es also zunächst, universellen Zugang zu einer Sammlung von Dokumenten zu erlauben. Webseiten lieferten reine Informationen. Da diese als Dokumente angesehen wurden, war eine Nutzerinteraktion mit der Webseite nicht möglich. Allerdings konnten Hyperlinks Webseiten miteinander verbinden. Dies war das wichtigste Merkmal.

|  |
| --- |
| Erkenntnisse aus Grafik |
| Die PC-Ära begann in den 1980er Jahren. Dabei wurde dieser nur als Dekstop-Version angesehen. Datenbanken, Ordner und deren Strukturen wurden entwickelt..  Daraufhin folgte das Web 1.0, was klassisch als World Wide Web bezeichnet wird.  Das Web 2.0, auch als Social Web bezeichnet, begann um die Jahrtausendwende. Revolutionär war, dass nach bestimmten Wörtern gesucht werden konnte. Wir befinden uns aktuell in der Ära des Web 3.0, dem sogenannten Semantic Web. Dieses wird vor allem durch das Tagging-Prinzip gekennzeichnet. Ein nächster Schritt, der mehr und mehr Form annimmt, ist die Suche durch natürliche Sprache. Die semantische Suche und das automatisierte Schlussfolgern gehen bereits in das Web 4.0 ein, was in den nächsten Jahren mehr und mehr Einfluss nehmen wird. Dieses wird auch als „The Intelligent Web“ gekennzeichnet. |

## Wissensorganisationssysteme am Beispiel von Tagging

Im Folgenden werden Begrifflichkeiten zum Themenschwerpunkt „Tag“ erläutert. Tagging bezieht sich hierbei auf Tags im Social Semantic Web.

### Definition

<http://hpl.hp.com/research/idl/papers/tags/tags.pdf> (!)

Tagging wird vor allem im Bereich der Wissensorganisation (siehe Wissensmanagement) eingesetzt und bildet einen elementaren Bestandteil. ..

Was sind Tags?

Tags etikettieren, Verschlagwortung

Stichwort Metadaten

[Siehe BA zum Thema Definition von Tags]

Tags können mit ihren beschreibenden Inhalten nach […] in sieben unterschiedliche Kategorien eingeteilt werden:

Identifying What (or Who) it is about – Tags, die den (groben) Inhalt beschreiben, über wen oder was es sich handelt (z.B. Enterprise IoT).

Identifying What it Is – Tags, die den Objekttyp beschreiben (z.B. Buch).

Identifying Who Owns it – Tags, die den Verfasser benennen (z.B. Dirk Slama & Frank Puhlmann).

Refining Categories – Tags, die allein noch keine Aussagekraft haben und erst in Kombination mit weiteren Tags einen Sinn ergeben (z.B. „50 Billion Connected Devices Identifying“: Ohne die Maßangabe kann aus der Zahl kein eindeutiger Sinn erschlossen werden.

Qualities or Characteristics – Tags, die die Meinung des Verfassers bezüglich des Objektes wiederspiegeln (z.B. falsch oder inspirierend)

Self Reference – Tags, die den Verfasser in Beziehung zum getaggten Objekt stellen (z.B. „meinUrlaub“)

Task Organizing – Tags, die Aufgaben beschreiben (z.B. „muss noch gegengelesen werden“)

(vgl. The Structure of Collaborative Tagging Systems Seite 5)

Was bedeutet Tagging?

Mit Tagging kann auf verschiedene Inhalte via eines Indexverfahrens referenziert werden. Inhalte sind beispielsweise Webseiten, verschiedene Dateiformate im Musik- und Bildbereich,… Das Tagging im eigentlich Sinn verfolgt zunächst keine logischen Relationen. Es wird eingesetzt, um etwas zu beschreiben. Erst in einem weiteren Schritt kann eine Ontologie eingebunden werden.

Social Bookmarks share ressources, Tags define ressources

### Automatisiertes Tagging

Beim automatischen Tagging werden Tags nicht manuell zu einem Inhalt hinzufügt, sondern durch einen bestimmten Algorithmus generiert. „[Die] Verfahren zur Extraktion von Stichworten, Schlagworten oder Schlüsselphrasen bedienen sich statistischer und computer-linguistischer Verfahren und stammen aus dem Umfeld der automatischen Sprachverarbeitung und der Künstlichen Intelligenz“ (siehe Social Semantic Web-Buch Seite 230).

Dieses Verfahren wird vor allem in Bereichen der Medientypen wie Textdokumente, Audio und Grafiken angewandt.

… Hier noch etwas zu Texten

Im Musikbereich können Metadaten wie Titel, Künstler, Album usw. durch Tools wie „Tag&Rename“, „MediaMonkey“ oder „TuneUp“ angepasst oder neu hinzugefügt werden. Dies geschieht [vermutlich] dadurch, dass der Mediastream abgehört und mit einer großen Datenbank verglichen wird, in der die Tags bereits abgespeichert wurden. [// Ist das Wissen hier überhaupt notwendig????]

Ein anderer Bereich ist im Grafik-Bereich das Tagging oder Taggen von Bildern. Dieses Gebiet befindet sich allerdings aktuell noch in der Erprobungsphase. So hat beispielsweise der Fotodienst flickr im Mai 2015 ein Bilderkennungssystem eingeführt, welches automatisch Tags an Bilder anfügt. Auch hier wird ein bestimmter Algorithmus angewandt, um Schemata auf einem Bild erkennen und bestimmten Tags zuordnen zu können. Allerdings zeigen sich hierbei die Grenzen des automatischen Taggings.

Da die Erkennung nur nach logischen Algorithmen erfolgt, fehlen in seltenen Fällen „zwischenmenschliche Informationen“. Diese können allerdings verheerende Folgen haben. So wurde beispielsweise ein Bild vom Konzentrationslager Dachau auf flickr automatisch mit den Metadaten/ Begriffen „Sport“ und „Klettergerüst“ getagt, was eine öffentliche Empörung mit sich brachte (siehe http://www.theguardian.com/technology/2015/may/20/flickr-complaints-offensive-auto-tagging-photos/). Diese fehlerhaften Tags wurden daraufhin manuell entfernt. Dies ist für das System notwendig. „If you delete an incorrect tag, our algorithm learns from that mistake and will perform better in the future. The tagging process is completely automated“. Der Algorithmus lernt demnach vom manuellen Handeln und ist davon abhängig. Auch wenn der Taggingprozess laut Flickr „komplett automatisch“ funktionieren soll. Ein weiterer Kritikpunkt ist laut Usern, dass automatische Tags nicht spezifisch genug sind.

An diesem Beispiel wird deutlich, dass automatisches Tagging das Tagging-Prinzip erleichtern kann, es aber zum einen eine gewisse Anlaufzeit benötigt und zum anderen von der Interaktion einer großen Anzahl von Nutzern abhängig ist. Diese Interaktion wird als „Social Tagging“ im nachfolgenden Kapitel beschrieben.

### Manuelles Tagging

Manuelles Tagging bedeutet, dass Nutzer/ Personen Tags manuell zu einem Inhalt hinzufügen. Ein großes Beispiel dafür ist Twitter. Dabei werden sogenannte Hashtags verwendet. Dabei setzt sich ein #-Zeichen mit einer knappen Beschreibung, bzw. Benennung einzelner Wörter. Zum Beispiel: #InternetOfThings .

🡪 Sagen, welche Art von Tags da benutzt werden (siehe BA)

🡪 Evtl eher als Anfang vom Social Tagging

#### Tagging durch Experten

#### Tagging durch den Endnutzer

### Social Tagging / Folksonomy

Auch bekannt als Folksonomies, collaborative tagging, social classification, social indexing, social tagging

<https://kommitblog.wordpress.com/2013/11/28/social-bookmarking-tagging-wissen-gemeinsam-sammeln-und-nutzen/>

<http://www.vwh-verlag.de/vwh/wp-content/uploads/2010/08/titelei_frohner.pdf>

<http://einsiedlerkrepsecm.blogspot.de/2013/02/taxonomien-und-folksonomie.html> + <https://books.google.de/books?id=md_JBAAAQBAJ&pg=PA92&dq=folksonomy&hl=de&sa=X&ei=o0FsVZS1AuPFygPhrYOQAw&ved=0CEEQ6AEwBA#v=onepage&q=folksonomy&f=false>

Social Tagging bezeichnet…

Der Begriff “Folksonomy” setzt sich aus den englischen Wörtern “Folk” (der Endbenutzer) und Taxonomy (siehe 2.1.3.1.1) zusammen. Es beinhaltet: „das kollektive Wissen der ‚normalen‘ Nutzer vernetzter Internet PCs […]. Im Gegensatz zur Taxonomie, üblicherweise definiert von Experten, bezieht sich Folksonomy auf von Endbenutzern definierte Spezifikationen, Klassifizierungen und Organisation von Informationen […] in umgangssprachlich verständlichen Ausdrucksformen“ (s. Bausteine der WI, S. 236)..

### Gegenüberstellung automatisiertes und manuelles Tagging

Manuelles Tagging in Form des Social Tagging bietet einige Vorteile. So wird es nicht nur von Experten durchgeführt, sondern ist für jeden Nutzer frei verfügbar. Durch diese Community ist es somit möglich, immer mehr und genauere Tags für einen Inhalt zu erstellen. Durch die persönliche Erstellung und Bewertung der Tags können noch präzisiere Angaben erstellt werden. Dies wirkt dem gravierenden Nachteil des Manuellen Taggings entgegen, dem diese soziale Komponente fehlt.

Ein großer Nachteil von Social Tagging ist das Problem der mehrfachen Verschlagwortung. Zum einen kann die Verwendung von Synonymen und Homonymen für Verwirrung sorgen.

Synonyme sind … [korrekte Definition hier einfügen]. Homonyme hingegen sind Wörter mit gleicher Schreibweise und Aussprache, aber unterschiedlicher Bedeutung.

<http://www.przepiorka.de/archiv/tagging-schlagwoerter/>

<https://de.onpage.org/wiki/Social_Tagging>

Nach <http://www.wissensgarten.com/wordpress/wp-content/uploads/ontologie_modelle.pdf> :

Falsche Schreibweise

Homonymie

Synonymie

Mehrsprachigkeit

Unterschiedliche Abstraktionsebenen

### Anwendungsbereiche

Tagging wird vor allem in Social Software eingesetzt. Dies beschreibt Software, Websites, etc., welche ihr Wissen durch die Zuarbeit ihrer Nutzer generieren und aufbauen. Beispiele hierfür sind:

Wikis („eine Umgebung, die Erstellung, Editierung und Update [einschließlich Entfernung] von Webseiten in einer collaborativen Umgebung durch e-Collaboration Mechanismen ermöglicht, ohne Login oder Registrierung“ [siehe Bausteine WI, S. 237])

Blogs („ein Web-basiertes persönliches Journal, adressiert und offen für alle oder eine spezifische Interessengruppe“ [siehe Bausteine WI, S. 239])

Social Bookmarking

(RSS-)Feeds („eine Familie automatisch arbeitender Protokollformate zur Veröffentlichung und Verteilung kontinuierlich aktualisierter Inhalte von Web-Seiten. Die Nutzung umfasst eine Web-basierte Datenquelle [RSS-Feed] und eine Empfängerseite [Feed Reader]“ [siehe Bausteine WI, S. 245])

Podcasts („ein Verfahren, beliebige Dateiarten, normalerweise Audio und Video, über das Internet zum eigenständigen Abspielen auf mobilen Geräten automatisiert zu verteilen“ [siehe Bausteine WI, S. 243])

Microblogs

Mashups

Social Networks

Auch in anderen Bereichen wie in Wissensdatenbanken und bei einer (Volltext-)Suche kommen Tags zum Einsatz. Dabei wird beispielsweise ausgewertet, wie oft ein Tag angeklickt wurde, welcher dementsprechend eine höhere Bedeutung erhält. Ebenso können Beziehungen zwischen Objekten und Tags hergestellt werden und je nach Relation zueinander bei einer Suche, bzw. in einer Wissensdatenbank genutzt werden.

Wissensdatenbanken

|  |
| --- |
| <http://de.wikipedia.org/wiki/Wissensdatenbank>  Vor allem die Hauptaufgaben:  “ Wissen anderen Personen zur Verfügung zu stellen,  Ressourcen schnell zu finden,  Informationen aller Art zu sammeln, zu kommentieren, zu verknüpfen und zu verwerten.  Information selektiv darzustellen (Vermeidung von Überhäufung mit nicht erwünschten Streuinformationen)  Informationen möglichst exakt (dem natürlichen Sachverhalt entsprechend), selbsterklärend (d. h. geometrisch im 3D-Bild) und ebenso stringent (d. h. durch Formel) darzustellen.” |

## Wissenssuche mithilfe semantischer Werkzeuge

Wie kann vorhandenes Wissen gefunden werden? In Zeiten vor dem World Wide Web gaben beispielsweise Enzyklopädien Antworten auf aufkommende Fragen. Heutzutage wird Wissen zum Großteil im Internet gesucht. Dieses basiert auf dem Semantic Web, was nachfolgend vorgestellt wird. Daraufhin wird genauer auf die semantische Suche eingegangen.

### (Web 3.0)

… (aus Wiki) Web 3.0 beinhaltet alle Konzepte des Web 3.0, wobei dabei noch die Konzepte des Semantic Web hinzukommen.

### Semantic Web

Die Semantik ist ein Teilgebiet der Sprachtheorie. Sie beschreibt „Beziehungen zwischen sprachlichen Zeichen (Wörter, Sätze) und ihrer Bedeutung sowie zwischen Zeichen und Bezeichnetem“ (Wirtschaftslexikon[[1]](#footnote-1)). Evolution

|  |
| --- |
| “The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation” |

Das Semantic Web wiederum wird wie folgt definiert: “The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation”[[2]](#footnote-2).

|  |
| --- |
| What is the Semantic Web?  The **Semantic Web** provides a common framework that allows **data** to be shared and reused across application, enterprise, and community boundaries. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. It is based on the Resource Description Framework ( [RDF](http://www.w3.org/RDF/)). See also the separate [FAQ](http://www.w3.org/2001/sw/SW-FAQ) for further information. |
| http://www.w3.org/2001/sw/ |

Durch das Semantic Web werden die Kernprozesse von Wissen (auch als „innere Prozesse“ bezeichnet) um Wissensbewertung und Wissensziele erweitert, wie in Abbildung 8 dargestellt wird.

|  |
| --- |
| Das Semantic Web ist eine ergänzende Schicht zum WWW und erweitert dies um die Dimension der Bedeutung.  Durch RDF (Subjekt – Verb – Objekt) werden dem Computer Dinge erklärt  Ontologien = (Theorien über die Wirklichkeit) spielen hierbei eine Rolle |

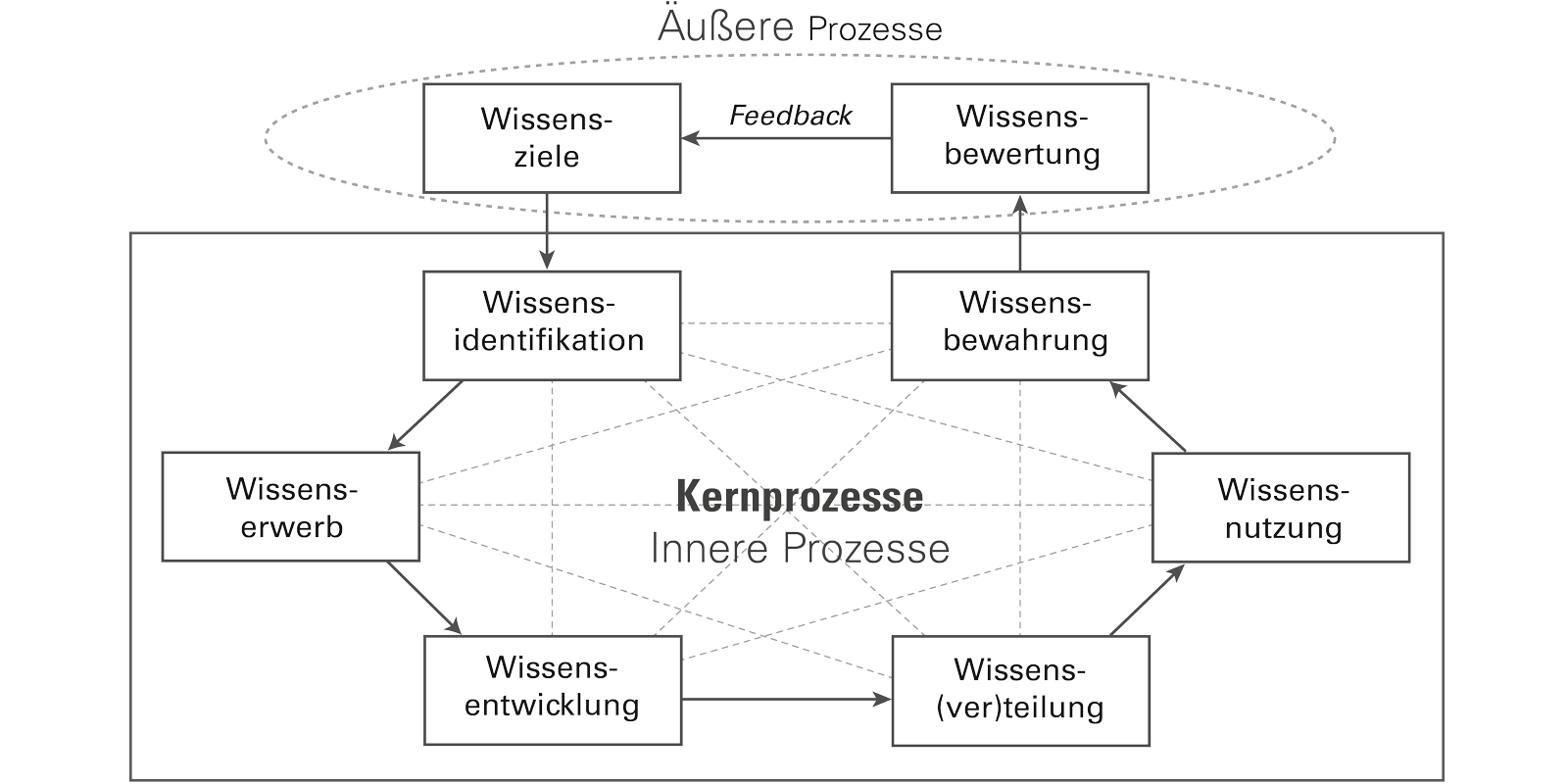


Abbildung - Bausteine des Wissensmanagements

Die als „Äußere Prozesse“ bezeichneten Bausteine stellen in diesem Modell Beginn und Ende des Wissensmanagements dar. Die Durchführung des gesamten Zyklus‘ erfolgt periodisch.

|  |  |
| --- | --- |
| Die Wissensbewertung dient als Qualitäts- sowie Erfolgskontrolle der benannten Zielvereinbarungen (vgl. Probst et al. 2010:31). Darüberhinaus können die bestehenden Strategien anhand der ermittelten Wissensindikatoren assimiliert werden. „Wissensbewertung ist eine essentielle Voraussetzung zur Einschätzung der Effizienz von Wissensmanagement. Sie gibt Auskunft darüber, ob Wissensziele angemessen formuliert und Wissensmanagement-Maßnahmen erfolgreich durchgeführt werden.“ (Probst et al. 2010:234)   |  | | --- | | Bei Probst richtig zitieren (Bausteine des Wissensmanagement) | |

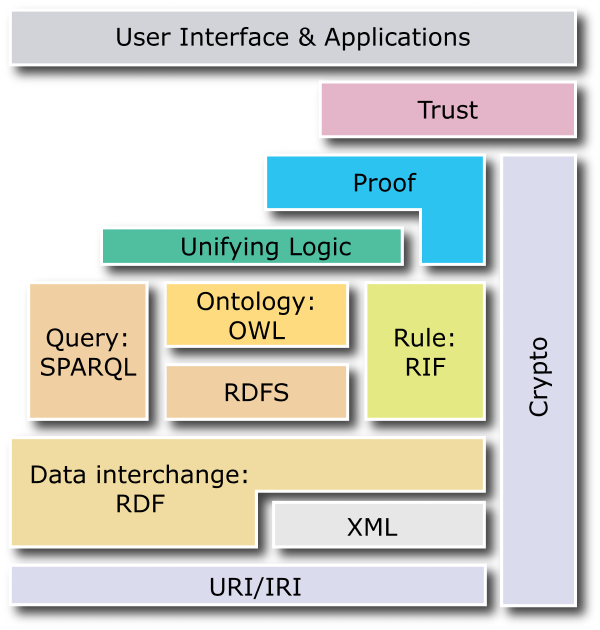


Abbildung Semantic Web Layer Cake nach W3L[[3]](#footnote-3)

### Semantische Suche

#### (Semantische Technologien) Technologische Verfahren zur Suchanfragenverarbeitung

Nach /cite{D.Lewandowski.2011} werden folgende technologische Verfahren zur Suchanfragenverarbeitung genutzt:

[**ZUSATZINFORMATION**](file:///C:\Users\gic1be\Dropbox\Master%20Thesis\Geschriebenes\Zusatzinformation%20Suchanfragenbearbeitung.docx)

#### Optimierungsmöglichkeiten einer Suchanfrage

Es besteht die Möglichkeit, Suchanfragen auf verschiedene Arten zu unterstützen. Welche Unterstützungsoptionen es gibt und welche aktiv in verschiedenen Suchmaschinen genutzt werden, wurde in einer Untersuchung festgestellt. Die Ergebnisse können aus Abbildung 10 entnommen werden.

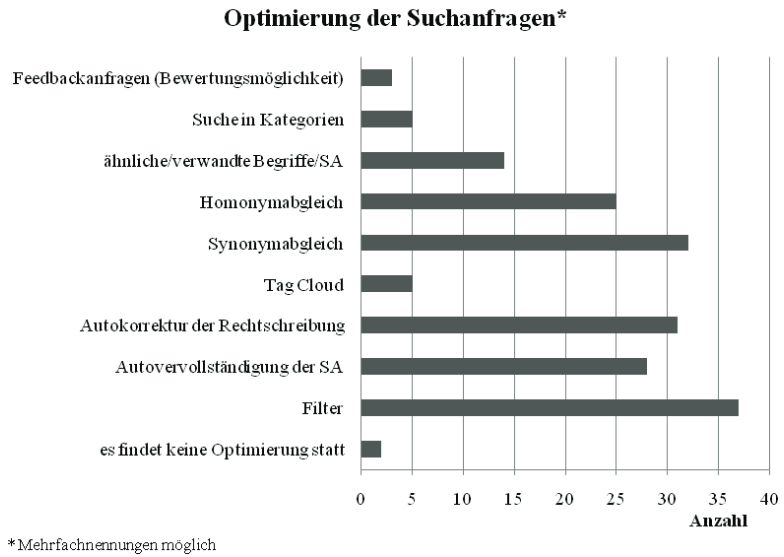


Abbildung Untersuchung von Optionen zur Optimierung von Suchanfragen nach D.Lewandowski.2011 S 92

In der Untersuchung wurden die nachfolgenden Optionen zur Optimierung von Suchanfragen benannt:

Feedbackanfragen (Bewertungsmöglichkeit)

Bei Feedbackanfragen wird der Nutzer gebeten, eine meist kurze Rückmeldung zu geben, ob das Suchergebnis seinen Erwartungen entsprochen hat.

Suche in Kategorien

Die Suche in Kategorien ermöglicht es, die Suchergebnisse einzugrenzen, indem nach Themenschwerpunkten gefiltert werden kann.

Ähnliche und / oder verwandte Begriffe der Suchanfrage

Es werden ähnliche und / oder verwandte Begriffe aus der Suchanfrage vorgeschlagen. Diese befinden sich meist in einer gemeinsamen Begriffsgruppe oder –kategorie.

Homonymabgleich

Homonyme sind Wörter mit gleicher Schreibweise und Aussprache, aber unterschiedlicher Bedeutung. So wird der User beispielsweise gefragt, ob er mit dem Suchbegriff „Golf“ ein Automodell oder eine Sportart meint.

Synonymabgleich

… ToDo

TagCloud

Eine Sammlung verschiedener Begriffe, die zur Suchanfrage passen. Auf dieses Verfahren wird im Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** genauer eingegangen.

Autokorrektur der Rechtschreibung

Bei falscher Schreibweise des Begriffes schlägt die Software eine Korrektur vor. Diese hängt unter anderem davon ab, ob der korrigierte Begriff mehr Treffer erzielen würde. Zudem wird die Datenbankbasis nach Begriffen mit ähnlicher Schreibweise durchsucht.

Autovervollständigung der Suchanfrage

Das System gleicht die Suchanfrage mit der Datenbank auf Einstimmigkeiten (?) ab und schlägt eine Ergänzung vor (zum Beispiel bei der Eingabe des Begriffes „Golf“ Ergänzungen wie „VW Golf“ oder „Golf Handicap“).

Filter

Der Nutzer wählt eine oder mehrere Kriterien aus, nach denen gefiltert werden soll. Meist wird die Suchanfrage erst durch einen Klick auf einen Button (zum Beispiel: „Filter anwenden!“) aktiviert.

## Das Internet der Dinge als Resultat des Web 3.0 (evtl. muss das hoch)

Das Internet der Dinge, auch als IoT (Internet of Things) abgekürzt, wird teilweise gleichbedeutend mit dem Begriff Web 3.0 verwendet. Es beschreibt …

ist neben

### Definition & Aktualität des Begriffs

Definition…

Eine der größten Herausforderungen von IoT ist es, sowohl die Bereiche des „Machine Camp“ als auch des „Internet Camps“ mit ihren Bedürfnissen abzudecken [Overture]. Sie werden nicht nur von kulturellen Differenzen beeinflusst, sondern auch von technischen Restriktionen. + Standards und Regularien

Neue Flexibilität: Z.B. in dem für 2Mio. Autos nur ein Softwareupdate kommt und sie nicht alle Autos zurück rufen müssen (Stichwort digitalisierte physische Produkte)

Die Grundlage des Internet der Dinge bildet die Digitalisierung der physischen Welt.

### Ignite Projektmethodik

…

Bedürfnisse von PM:

IoT Projektmanager stehen vor verschiedensten Herausforderungen. Zum einen sind es Dinge, die jeder Projektmanager, unabhängig seines Einsatzgebietes, kennt:

…

…

…

Im Bereich IoT kommen allerdings noch weitere Themengebiete hinzu. So steht ein IoT Projektmanager vor der Herausforderung, nicht nur ein Team von fünf bis zehn Personen zu leiten, sondern mehrere Teams. Diese können an verschiedenen Standorten positioniert sein, wobei auch einzelne Personen an einem anderen Ort sein können. Hinzu kommen mehrere Sprachen und Zeitzonen.

Big Data

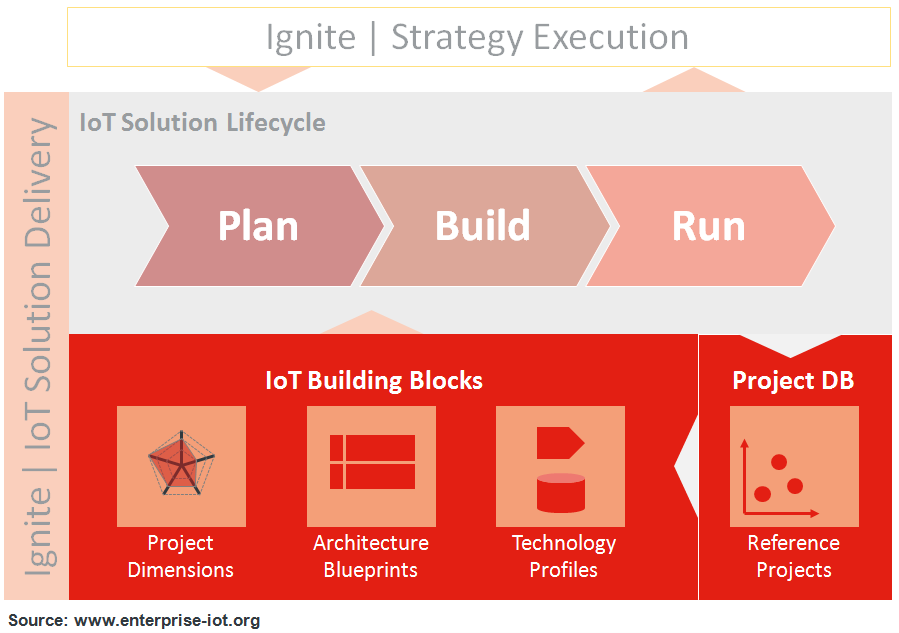
Unbekanntes Terrain evtl

…

Enterprise IoT:

In Ignite wurde der Lebenszyklus der IoT Lösung mit den drei Kernphasen „Plan“, „Build“ und „Run“ beschrieben. Die nachfolgend benannten „IoT Building Blocks“ sind dabei eine noch detaillierte Beschreibung der Artefakttypen, welche aus dem Lebenszyklus stammen. Exakt stammen sie aus der Planungsphase.

Projektdimensionen bilden einen Maßstab für Projektanforderungen. Die Architektur beschreibt, aufbauend auf bereits bekannte Architekturen, welche noch für eine IoT Lösung benötigt werden. Das Technologieprofil benennt typische Technologien, die für ein IoT Projekt notwendig sind.



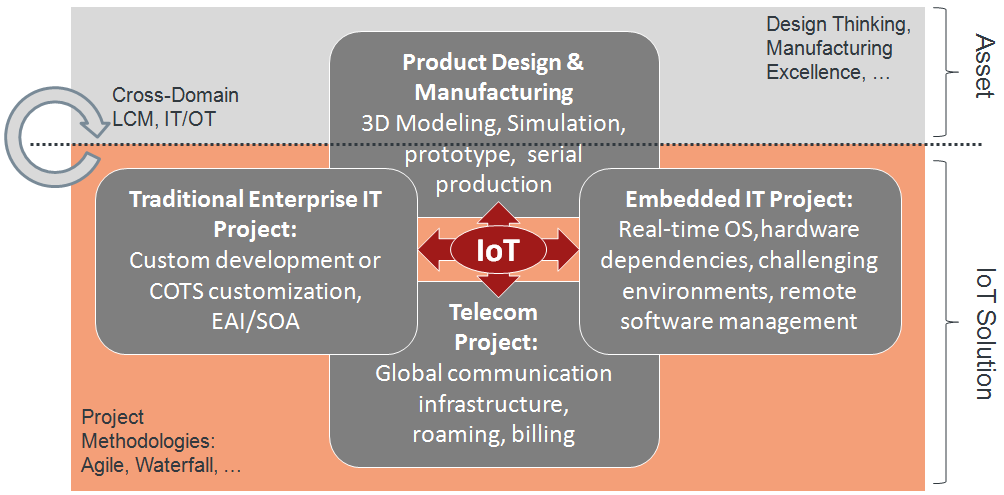


Abbildung Sichtweise von Projektmanagern (<http://enterprise-iot.org/book/enterprise-iot/part-ii-igniteiot-methodology/igniteiot-solution-delivery/>) 🡪 mit einbauen (Meeting 20150611)

# Entwicklung eines Wissensorganisationssystems

In der Entwicklungsphase wird das zukünftige Wissensorganisationssystem modelliert. Daraufhin werden mögliche Umsetzungsszenarien analysiert, um anschließend den Entwurf der Datenbank und der Softwarearchitektur vorzustellen.

## Modellierung des Wissensorganisationssystems

Die Entwurfsphase wird in Anlehnung an die Phasen der Wissensmodellierung nach (Blumauer2009, S. 137 ff) durchgeführt. Wie bereits in Kapitel 2.1.4 Wissensmodellierung beschrieben, beginnt die Modellierung, indem Rahmen, Zweck und Umfang eingegrenzt werden. Daraufhin werden Wörter gesammelt, wesentliche Begriffe identifiziert sowie kontrolliert und Klassen gebildet. Dies erfolgt in diesem Falle automatisch mithilfe einer Software. In der fünften Phase werden Relationen festgelegt, wodurch automatisiert die Art des Wissensorganisationssystem zugewiesen wird. Es folgt eine Vernetzung der Klassen mittels Relationen sowie die Definition der Eigenschaften von Klassen. Zuletzt erfolgt eine Vervollständigung durch die Individualebene in Form der Autorin.

### Eingrenzung von Rahmen, Zweck und Umfang

Um Rahmen, Zweck und Umfang eingrenzen zu können, werden nachfolgende Fragen zu den Themenfeldern Kontext, Benutzer / Nutzeranalyse, Inhalt und Vokabular beantwortet:

Kontext:

|  |  |
| --- | --- |
| Warum wird das Wissensorganisationssystem (WOS) erstellt? | Das WOS basiert auf der Idee, Wissen zu Projekten zu modellieren und dieses semantisch zu suchen. Dementsprechend sollen bereits evaluierte IoT Projekte miteinander verglichen werden können. Es soll ebenso möglich sein, nach Projekten bzw. einzelnen Eigenschaften davon zu suchen. Ebenso sollen neue Projekte angelegt werden können. |
| Welche expliziten und impliziten Ziele werden damit verfolgt? | Explizit:  IoT Projekte miteinander vergleichen  Hilfestellung (in Bezug auf eigene Projekte) für IoT Projektleiter geben  Mehr Informationen über IoT Projekte erhalten und diese verarbeiten  Implizit:  Projektleiter motivieren, am Prinzip der offenen Dokumentation (Open Documentation) teilzunehmen, um so sein eigenes Wissen zu teilen und dadurch eine gemeinsame Wissensbasis aufzubauen. |
| In welchem Zeitraum soll das Wissensorganisationssystem erstellt werden? | Das WOS soll prototypisch während des Zeitraums der vorliegenden Master Thesis geschaffen werden. |
| Welche technologische Infrastruktur steht für die Umsetzung des WOS zur Verfügung? Welche Vorgaben ergeben sich daraus für ein Tool zur Modellierung und für die Integration des WOS? | Aufgrund der prototypischen Implementierung wird das System zunächst auf einer lokalen Umgebung aufgesetzt. Zukunftsbetrachtend wird es notwendig sein, die Umgebung auf einem Server umzuziehen, um so den Zugriff von außen zu ermöglichen.  Das Tool soll die Möglichkeit bieten, Projekte zu finden (und vergleichbar zu machen). Dies soll über die reine Volltextsuche hinaus gehen und semantische Ansätze behandeln.  Die Umgebung soll in einem Webspace einbindbar sein. Dies bedeutet, dass potentielle Nutzer eine Seite aufrufen, um nach Projekten suchen können. |
| Welche vom Auftraggeber vorgegebene Rahmenbedingungen sind weite[r] zu beachten? | Die Datenbank soll für Interessierte frei verfügbar sein, was das Prinzip der offenen Dokumentation von Enterprise IoT wiederspiegelt. Daher soll möglichst freie Software verwendet werden. Ausnahmen können hierbei Produkte von Bosch SI bilden. Im speziellen kann es sich dabei um Software handeln, die selbst von der Firma entwickelt und vertrieben wird. |

Tabelle - Fragen zum Kontext des WOS (Struktur der Fragestellung nach SSW2009, S. 139)

Benutzer / Nutzeranalyse

|  |  |
| --- | --- |
| Wer sind die potentiellen Nutzer des Systems? | Potentielle Nutzer des Systems sind IoT Projektmanager.  Es werden Projektmanager angesprochen, die sich neu in das Themenfeld IoT einarbeiten und nach Wissen suchen. Ebenso richtet es sich an Projektmanager, die bereits Erfahrung in diesem Bereich haben und ihr Wissen teilen. |
| Welche (unterschiedlichen) Benutzergruppen sollen das WOS benutzen? […] Welche Daten […] liegen diesen Benutzergruppen vor? | Benutzergruppen wissensinteressierte IoT Projektmanager, die bereits Erfahrung im Themengebiet haben oder sich neu einarbeiten. Der letzten Gruppe liegen wenig bis keine Daten vor, erfahrene kennen Daten über eigene IoT Projekte. |

Tabelle - Fragen zu den Benutzern des WOS (Struktur der Fragestellung nach SSW2009, S. 139)

Ebenso müssen die Bedürfnisse der Nutzer bedacht werden. Die Tätigkeit als Projektmanager ist grundsätzlich mit dem Einhalten verschiedener Terminen und Meilensteinen verbunden. Da diese Priorität haben, darf die Verwendung des Tools nicht zu viel Einarbeitungszeit benötigen. Dementsprechend ist es bedeutend, dass alle wichtigen Informationen auf einen Blick abrufbar sind. Dies wäre in diesem Fall beispielsweise durch die Verwendung von Spider-Diagrammen möglich, welche bereits in der Excel vorhanden sind. Ein anderes Beispiel ist eine Tag Cloud. Auch eine erweiterte Suche könnte hier Abhilfe schaffen. Wichtig ist, dass ein Nutzer nicht gezwungen wird, sich vor Verwendung des Systems zu registrieren.

Aufgrund der Zeitknappheit muss desweiteren ein Anreiz gegeben werden, Projekte selbst einzupflegen. Auch hier könnte eine Tag Cloud oder die Aussicht auf eine Generierung eines persönlichen Spider Diagramms unterstützend wirken.

Inhalt

|  |  |
| --- | --- |
| Welche Arten von Dokumenten sollen erschlossen werden (Struktur der Dokumente und Dateitypen)? | Die Arten von Dokumenten können in Faktoren unterschieden werden.  Harte Faktoren bilden Projektdaten, die bereits strukturiert in einem Excel-Sheet abgespeichert sind. Hinzukommen kommen Daten über Projekte, die in mehreren natürlich sprachlichen Texten festgehalten sind.  Weiche Faktoren sind einzelne Wörter, die die Projekte beschreiben sollen und miteinander verbunden sein können. |
| Welchen Umfang haben die zu erschließenden Dokumente (Anzahl, Länge der Dokumente und Speicherkapazität)? | In der Excel-Tabelle befinden sich zu jedem der sieben Projekte gut 60 Informationen. Diese bauen teilweise aufeinander auf (beispielsweise als Zusammenfassung bestimmter Kennzahlen).  Weitere Projekten können nochmals jeweils 60 Informationen enthalten. Diese begrenzen sich auf Kennzahlen und Beschreibungen. Es werde keine Dateien im mp3-Format oder ähnlichem erhoben. |
| Wie schnell ist der Dokumentenbestand bisher gewachsen? Welches Wachstum des zu erschließenden Dokumentenbestandes ist zu erwarten? | Via Case Studies wurden Informationen über IoT Projekte erschlossen. Diese sind als initialer Bestand vorhanden. Weitere Informationen und Dokumente wurden seither nicht aufgenommen. |

Tabelle Fragen zum Inhalt des WOS (Struktur der Fragestellung nach SSW2009, S. 139)

Vokabular

|  |  |
| --- | --- |
| In welcher Form sind diese Dokumente bisher erschlossen? Gibt es bereits Metadaten (auch Tags) zu diesen Dokumenten? | Informationen zu den Projekten sind für sieben in einem Excel-Sheet abgelegt. Alle weiteren Informationen sind in Schriftform im Buch festgehalten und können online abgerufen werden. Metadaten und Tags sind bisher nicht vorhanden. |
| Was sind die inhaltlichen Schwerpunkte und Randgebiete dieser Dokumentenbasis? Welche Inhalte sollen nicht modelliert werden? | Der inhaltliche Schwerpunkt orientiert sich an den rund 60 Kriterien zu den einzelnen Projekten. Zusätzlich soll es möglich sein die Projekte mit Schlagworten versehen zu können. Diese beinhalten keine inhaltliche Grenzen. |
| Wie spezifisch soll das WOS werden? Wie allgemein oder spezifisch sollen die verwendeten Begriffe sein? | Da verwendete Begriffe von Nutzern erstellt und eingetragen werden sollen, gibt es keine Begrenzung. Sowohl allgemeine als auch spezifische Begrifflichkeiten sollen möglich sein. |
| Wie viele Begriffe werden voraussichtlich für die Beschreibung dieser Dokumente erforderlich sein (von – bis Schätzung)? | In der Höhe sind keine Grenzen gesetzt. Allerdings muss mindestens ein Begriff (eher mehr) das Projekt beschreiben können. |
| Welche Arten von Relationen sollen zwischen den Begriffen voraussichtlich hergestellt werden (gleichbedeutend, hierarchisch, assoziativ)? | Es sollen sowohl gleichbedeutende, hierarchische als auch assoziative Relationen hergestellt werden können. Dies basiert darauf, dass nach direkten Begriffen (in diesem Fall Projektnamen) gesucht werden können soll. Hinzukommen Schlagwörter, die klassifiziert sein können, aber nicht müssen. Da das Ziel ein semantisches Netz ist, wird zusätzlich eine assoziative Relation benötigt. |

Tabelle Fragen zum Vokabular des WOS (Struktur der Fragestellung nach SSW2009, S. 139f)

### Wortfindungen, Bilden von Relationen und Klassifizierungen

Wörter zu finden kann laut Literatur vor allem durch eine Recherche in verschiedene Lehrbüchern und Fachliteraturen erfolgen. Ebenso sollen bestehende Datenbanken durchsucht werden (vgl. SSW2009, S. 141). Dies ist in diesem Fall nicht möglich, da es diese Informationen noch nicht gibt. Allerdings können die vorhandenen Tabellen und die Struktur des Buches als Grundlage genutzt werden. Um Wörter aus dem Buch und vor allem aus dem Kapitel der Case Studies zu entnehmen, wird ein anderes Verfahren angewandt.

Wie bereits in Kapitel **Fehler! Verweisquelle konnte nicht gefunden werden.** beschrieben wurde, gibt es drei Möglichkeiten, eine Annotation von Ressourcen durchzuführen: Entweder durch den End-User, durch einen Experten oder durch eine vollständige Automatisierung des Verfahrens. Um ein sehr gutes Ergebnis zu erhalten, werden für den Prototypen alle Möglichkeiten angewandt.

Dies geschieht für die Automatisierung im ersten Schritt mittels der Software Open Calais. Durch diese Software ist es möglich, erste Wortfindungen, Relationen und Klassifizierungen automatisch zu generieren und somit das Verfahren des Auto-Tagging anzuwenden. Die Nutzung von Open Calias wird nachfolgend an einem Beispiel erläutert:

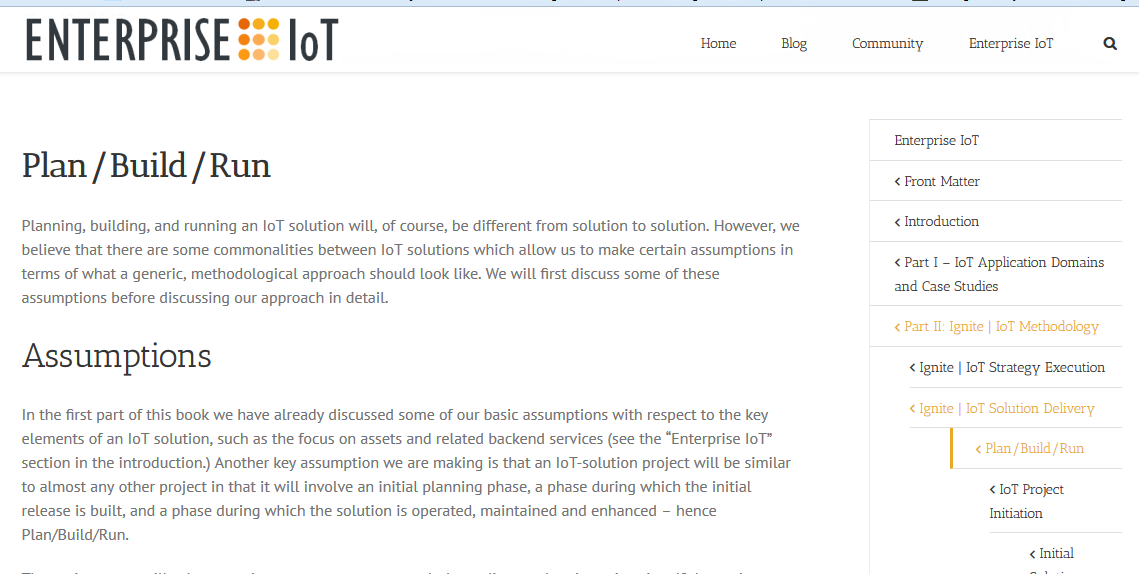


Abbildung - Auszug aus Beispieltext

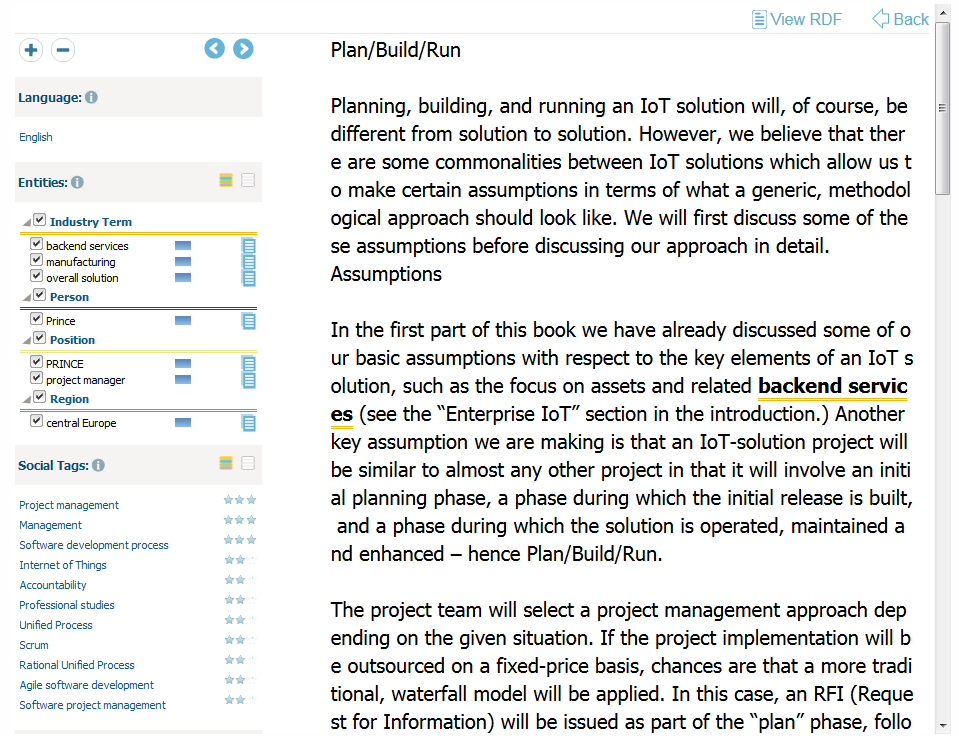


Abbildung - Ergebnis nach Analyse durch Open Calais

„[…] wird für die Datenbasis das Auto-Tagging-Prinzip genutzt. Hierfür wird die Software Open Calais verwendet. Open Calais generiert aus Volltexten Tags …“]

In Abbildung 12 ist ein Auszug eines Beispieltextes aus dem Buch Enterprise IoT zu erkennen. Es beinhaltet den Abschnitt „Plan/Build/Run“, welcher via Open Calais analysiert wird (siehe Abbildung 13). Sogenannte „Social Tags“ emulieren, wie ein echter Nutzer den Text getaggt haben könnte. Dafür wird der gesamte Text mit Wikipedia verglichen. Die Ergebnisse sind unter anderem „Projektmanagement“, „Management“ und „Softwareentwicklungsprozesse“. Dies entspricht dem tatsächlichen Inhalt.

Problematischer sind die Beziehungen (Entities). Dabei erkennt die Software zwar Backend Services, Manufacturing und Overall solution als Industriethemen, allerdings wird die Projektmanagementmethode PRINCE2 als Person & Position erkannt. Diese nicht korrekte Angabe resultiert daraus, dass im Buch die Methodik fälschlicherweise mit einem Lehrzeichen zwischen PRINCE und 2 angegeben wurde (die Eingabe eines korrigierten Textes ergab ein Ergebnis ohne PRINCE2). Dies ist im ersten Schritt der Sammlung von Tags und Beziehungen aber nicht zu folgenschwer, da im Verlauf der Nutzung des Prototypen User diesen Fehler erkennen und korrigieren sollen.

Durch die Verwendung der Software wird eine Wortsammlung vorgenommen. Die automatisierten Ergebnisse werden im Nachhinein von der Expertin bereinigt. Daraufhin wird das Social Tagging Prinzip angewandt. Bei diesem kann der End-User Tags hinzufügen und bereinigen, um so die Datenbasis zu verbessern.

Neben der Wortsammlung können auch erste Klasse gebildet werden. Essentielle Klassen sind nach der Expertin die nachfolgenden:

Es gibt die Klasse „Projects“. Diese beinhaltet Informationen wie Namen und Kennzahlen.

Jedes Projekt kann zudem Zusatzinformationen enthalten. Diese werden als Klasse „AddInformation“ benannt. Die zusätzlichen Informationen können unterschiedlicher Natur sein. So zum Beispiel Synonyme, Homonyme, Hyperlinks oder Tags. Die Klassifizierung kann erfolgen, ist aber keine Voraussetzung.

### Bilden von Relationen

Als Relationstypen werden nachfolgend definiert:

belongs\_to – Üblicherweise eine AddInformation, die mindestens einem Projekt gerichtet zugeordnet wird.

part\_of – Als Teil eines Ganzen ist es durch die part\_of-Beziehung möglich, eine Art von Klassen zu bilden.

same\_as – Diese Relation bildet Synonyme ab.

Bei den ersten beiden Relationen muss die Eltern-Kind-Beziehung klar sein, um so eine Hierarchie zu ermöglichen. Same\_as stellt keine Hierarchie dar und kann als diese auch nicht verwendet werden. Die Beziehungen untereinander sind in Abbildung 14 graphisch dargestellt.

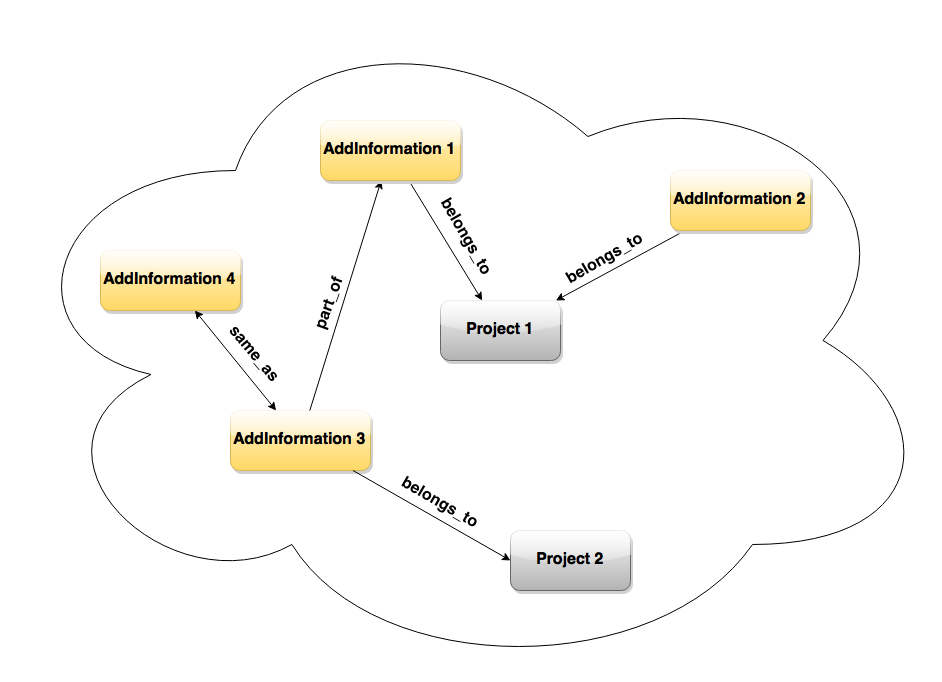


Abbildung - Relationen und Klassen

Aufgrund der vorhandenen Relationen kann nun definiert werden, welches WOS genutzt werden kann.

Eine Klassifikation als Relationstyp kann aufgrund ihrer reinen hierarchischen Beziehungen nicht angewandt werden. Ebenso schränkt ein Thesaurus zu sehr ein. Aufgrund der späteren manuellen Eingabe seitens der Benutzer ist ein semantisches Netz, bzw. eine Ontologie für diesen Fall notwendig. Für semantische Netze sind keine Standards verfügbar. Wohingegen es bei der Ontologie durch Bezeichnungsstandards, benannt durch W3C, möglich ist. Dabei werden RDF und OWL als mögliche Standards angeboten.

### Bilden von Klassifizierungen

Die Klasse Projects muss mindestens einen Projektnamen enthalten. Optional können alle benannten Eigenschaften aus dem Excel-Sheet einem Projekt zugeordnet werden (siehe Anhang). Projekte können nicht direkt miteinander verbunden werden.

Die Klasse AddInformation beinhaltet ein einzelnes Wort als Tag oder einen Hyperlink zu anderen Seiten. AddInformation kann eine direkte Relation zur Klasse Projects haben. Ebenso ist eine Relation zu anderen AddInformation möglich (siehe Kapitel 4.1.3).

### Vervollständigung

In der Phase der Vervollständigung wird die Datenbasis mittels Open Calias um die Annotation neuer Projekte erweitert. Die Anzahl der Klassen wird nicht erhöht. Ebenso werden keine weiteren Relationen hinzugenommen.

## Mögliche Umsetzungsszenarien

Mittels der zuvor definierten Kriterien und Eigenschaften für das Wissensorganisationssystem wird evaluiert, wie die Umsetzung realisiert werden kann. Dafür wird zunächst eine Marktanalyse durchgeführt. Dabei werden vorhandene Wissensorganisationssystem nach den benannten Kriterien in Kapitel 4.1 (Modellierung des Wissensorganisationssystems) evaluiert. Nach der Marktanalyse werden verschiedene Datenbankmodelle auf Eignung für das Wissensmanagementsystems untersucht. Ergebnis dessen ist eine Handlungsempfehlung.

### Marktanalyse

Der Markt von Wissensmanagementorganisationsystemen wächst stetig und bietet eine Vielzahl an möglichen Anwendungsszenarien. Um zu evaluieren, ob auf diesem Markt bereits eine Software vorhanden ist, die allen Ansprüchen aus Kapitel 4.1 gerecht wird, werden K.O.-Kriterien aufgestellt und Such-Kriterien benannt, um eine erste Selektion des Marktes durchführen zu können. Daraufhin werden Tools, die diese Anforderungen erfüllen, genauer untersucht.

#### Kriterien der Marktanalyse

Die K.O.-Kriterien wurden aus der Entwurfsphase 1 in Kapitel zusammen getragen und sind die nachfolgenden:

Freie Verfügbarkeit der Software

Möglichkeit zur Integration in eine Website

Schlankes Design

Freie Verfügbarkeit bedeutet, dass die Verwendung der Software Keine Kosten verursachen darf, weder bei der Erwerbung noch im Nachgang (z.B. nach einer gewissen Evaluierungsphase). Zudem muss auf das Tool online zugegriffen werden können. Dies setzt weiterhin voraus, dass es in eine Website integriert werden kann. Damit wird gesichert, dass sich der Inhalt im Kontext Ignite befindet und dementsprechend auf der Website eingebunden werden kann. Desweiteren ist ein schlankes Design Voraussetzung. Dies wird benötigt, um den späteren User eine einfache Handhabung zu garantieren. Dementsprechend sind ergänzende Eigenschaften möglich, finden jedoch keine positive Bewertung in der Marktanalyse.

Die Anbieter am Markt wurden nach folgenden Suchkriterien selektiert:

Software, die im Kurs „Wissensmanagement“ im Wintersemester 2014/2015 an der HTW Berlin untersucht wurde.

Anbieter, die über die Suchbegriffe „Semantic Search Open Source“ gefunden wurden.

Ergebnis dessen ist eine Liste von 13 Anbietern.

#### Ergebnis der Marktanalyse

Das Ergebnis der Marktanalyse ist dem Excel-Sheet im Anhang A zu entnehmen.

Es wird deutlich, dass vor allem das Kriterium der freien Verfügbarkeit nicht alle Anbieter erfüllen können. So fallen dadurch Marktführer wie TheBrain und Cmap (dieses ist nur für den Bildungsbereich und für die amerikanische Regierung frei verfügbar) aus der näheren Betrachtung raus. Die mögliche Integration in eine Website als zweites Kriterium können weitere Tools nicht anbieten, da viele nur für Desktopuser zur Verfügung stehen. Andere Tools können nur über die Webseite des Herstellers oder als eigenständige Websites auf dem Localhost aufgerufen werden. Die übriggebliebene Software unterscheidet sich in ihrem Umfang, der teilweise für den betrachteten Rahmen zu umfassend ist.

Alle Kriterien erfüllen konnten nur drei Anbieter, die nachfolgend genauer betrachtet werden: DeepaMehta, Open Semantic Search und Open Semantic Framework.

#### Detaillierte Betrachtung ausgewählter Software

Nachfolgend wird die angebotene Software von DeepaMehta, Open Semantic Search sowie Open Semantic Framework untersucht. Dabei wird sich auf das offizielle Angebot der jeweiligen Anbieter fokussiert, sowie auf die Realisierung der möglichen Umsetzung im Kontext der Master Thesis. Die Voraussetzungen zur Inbetriebnahme sind für jede Software ähnlich und beinhalten Programmiersprachen wie Java, Pyhton und PHP5, das Betriebssystem Linux sowie den Webserver Apache 2.

##### DeepaMehta

Die Softwareplattform DeepaMehta stellt Wissensmanagement visuell dar. Die Schnittstelle soll situationszentriert sein und sich vom klassischen Taxonomiegedanken unterscheiden. Dies geschieht dadurch, indem Topic Maps erstellt werden können. Eine automatische Zuweisung und Anordnung der erstellten Informationen erfolgt nicht. Je nach Berechtigungsstufe können sich Topic Maps passiv angesehen oder aktiv bearbeitet werden. Die Software nutzt Neo4j als Datenbank.

Eine Installation der Software erfolgt ohne Hindernisse. Die aktive Nutzung birgt allerdings verschiedene Probleme: Das Tool ist inkonsistent. So traten nicht verschuldete Java-Fehler auf, die ein komplettes Neuaufsetzen des Systems erforderten. Auch ein Test zur Verwendung von Homonymen konnte nicht erfolgreich abgeschlossen werden, da auch dieser zu Systemfehlern führte. Ein weiterer Kritikpunkt ist, dass kein Einblick in die Datenbank gewährt werden kann. So wird nicht deutlich, wie die zu Wissen generierten Daten gespeichert werden. Ebenso wäre ein hoher Anpassungsaufwand von Nöten, um über eine Volltextsuche hinaus eine semantische Suche einzubinden. So müsste beispielsweise mindestens ein PlugIn geschrieben werden. Für diesen Fall sind allerdings keine Dokumentationen vorhanden.

Da das System zwar die K.O.-Kriterien erfüllt, in seiner Anwendung aber schon beim Start Probleme hat, kann DeepaMehta nicht verwendet werden.

##### Open Semantic Search

Open Semantic Search ist eine Suchmaschinensoftware, die auf Apache Lucene / Solr basiert. Es wird unter anderem eine erweiterte Suchanwendung inklusive Volltextsuche und Tagging angeboten. Zudem ist die Software sowohl als Desktop-Version als auch als "Open Semantic Search Server“ angeboten. Es wird eine SQL-Datenbank verwendet.

Im Falle der Master Thesis wird die Server-Version benötigt. Die Installation konnte zwar abgeschlossen werden, eine Evaluation der Software ist aber nicht möglich. Grund dessen sind fehlende Komponenten im Installationspaket. Daher wurde Kontakt mit dem Support von Open Semantic Search aufgenommen. Dieser brachte die Aussage mit sich, dass alles – nicht wie auf der Homepage beschrieben – noch in der Entwicklung sei. Bezüglich der produktiven Verwendung wurde seitens des Herstellers darum gebeten, vorsichtig zu sein. Tagging sei zudem auch nur mit erhöhtem Implementierungsaufwand nötig.

Aufgrund der nicht vollständigen Installationsmöglichkeit und der Warnung des Supports wird Open Semantic Search nicht weiter in Betracht gezogen.

##### Open Semantic Framework

Open Semantic Framework nutzt nach eigener Aussage semantische Technologien für das Wissensmanagement. Dafür werden für das Tool verschiedene Erweiterungen angeboten, so zum Beispiel ein „Ontologies Management Tool“. Es wird über die Apache 2 Lizenz zur Verfügung gestellt und nutzt eine relationale Datenbank.

Die Installation von Open Semantic Framework kann trotz vorhandener Dokumentation nicht ohne Probleme umgesetzt werden. Mehrmaliges Neuaufsetzen der gesamten Software ist nötig, um eine lauffähiges System zu realisieren, was evaluiert werden kann. Grund dessen sind Fehlerquellen, die an verschiedenen Stellen der Installation auftreten. Nachdem diese manuell behoben werden, zeigt sich die Konfiguration als schwierig. Es treten verschiedene Rechteprobleme auf, obwohl sowohl auf dem getesteten System als auch innerhalb der Software der einzige User ein Administrator mit allen Rechten ist. Da dieses Problem nicht gelöst werden kann, kann die Software nicht evaluiert und somit nicht weiter betrachtet werden.

##### Zusammenfassung

Die evaluierte Software konnte im Testbetrieb nicht überzeugen. Installations- und Rechteprobleme an verschiedenen Stellen machten eine vollständige Evaluierung nicht möglich. Andere Probleme waren Fehler in der Software selbst, die eine spätere produktive Nutzung nicht ermöglichen.

Insgesamt kann festgehalten werden, dass der Markt an frei verfügbarer semantischer Software, die eine Suche unterstützen soll, derzeit keine Alternative bietet.

### Gegenüberstellung verschiedener Datenbankmodelle

Da in Kapitel 4.2.1 kein passendes System gefunden wurde, werden die Anforderungen aus Kapitel 4.1 selbst umgesetzt. Grundlage hierfür bildet eine Datenbank, die alle Informationen zum vorhandenen Wissen gespeichert hat und zukünftige Informationen mit aufnehmen soll. Allerdings muss auch hierbei differenziert werden, was die Anforderungen an das Tool sind und wie diese bestmöglich umgesetzt werden können. Daher wird zunächst eine mögliche Umsetzung mithilfe einer relationalen Datenbank mit mySQL vorgestellt. Daraufhin wird das Prinzip von NoSQL am Beispiel einer Graph Datenbank genauer betrachtet.

#### Relationale Datenbank am Beispiel von mySQL

Das Umsetzungsszenario sollte ursprünglich mithilfe einer relationalen Datenbank umgesetzt werden. Hierbei wurde sich aufgrund von Erfahrungswerten für mySQL entschieden.

Das UML-Diagramm in Abbildung 15 zeigt die mögliche Umsetzung, welche via MySQL Workbench modelliert und erstellt werden sollte.

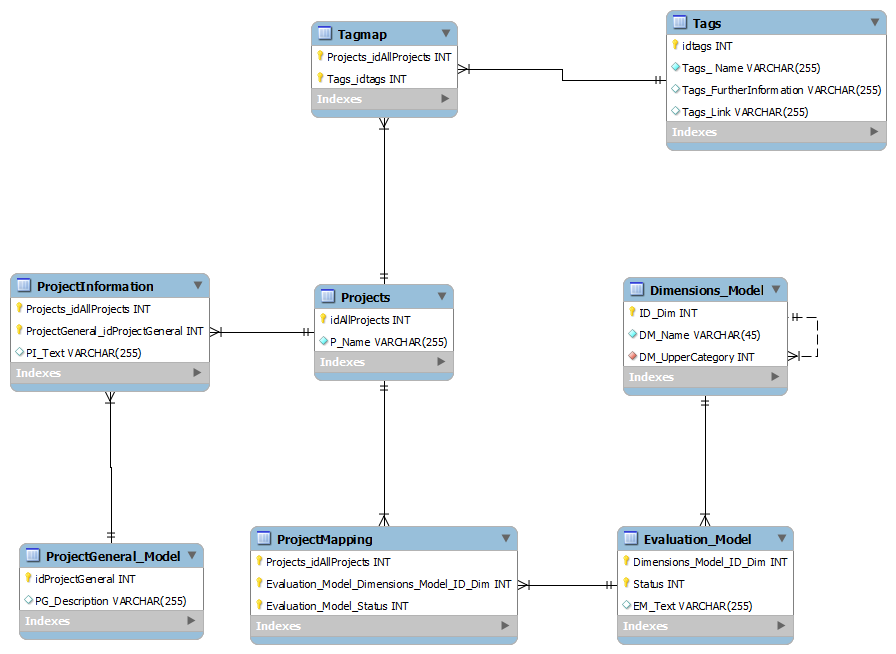


Abbildung Entwurf mithilfe einer mySQL-Datenbank

Um ein Projekt anzulegen, werden die unteren sechs Tabellen benötigt. Dimensions\_Model beinhaltet alle Dimensionensnamen inklusive Elternknoten, so wie es in der Tabelle angegeben ist. Evaluation\_Model enthält alle Auswahlmöglichkeiten als Text. So zum Beispiel, dass die Auswahl 1 in der Kategorie „Number of Assets“ den Wert „100s“ enthält. Im ProjectMapping sind diese Informationen für jedes einzelne Projekt und jede einzelne Zeile hinterlegt. Projects beinhaltet den Projektnamen und die jeweilige Projekt-ID. In ProjectInformation sind alle allgemeinen Angaben zu einem Projekt enthalten, wie beispielsweise der Interviewpartner. Die Klassifizierung dessen erfolgt in ProjectGeneral\_Model.

Projekte können sich so leicht anlegen lassen. Eine Abfrage nach einem Projekt mit allen Eigenschaften, wie sie in der Excel-Tabelle vorhanden sind, ist allerdings nur doch mehrere Joins möglich. Dies kann allerdings mithilfe einer View koordiniert werden.

Neben den Projekten werden AddInformation benötigt. Diese werden in diesem Beispiel als „Tag“ benannt und sind in der Tabelle Tags hinterlegt. Eine Zuordnung zu einem oder mehreren Projekten ist ind er Tabelle TagMap festgehalten.

Auch Tags können nach diesem Prinzip leicht angelegt und direkt an Projekte gebunden werden.

Eine Abfrage, die in diesem Fall allerdings nicht realisiert werden kann, ist die Umgebungssuche. Diese ist für eine semantische Suche notwendig, um zu ermitteln, welche Projekte beispielsweise gleiche oder ähnliche Tags enthalten, um so eine mögliche kurze Distanz in einem semantischen Netz zu entdecken. In diesem Fall wären Zusatzinformationen und mehrere Join-Abfragen notwendig. Dies entspricht allerdings nicht dem Prinzip der Semantik . Eine Alternative dazu bieten Graph Datenbanken.

#### Graph Datenbank am Beispiel von Neo4j

[**ZUSATZINFORMATION**](file:///C:\Users\gic1be\Dropbox\Master%20Thesis\Geschriebenes\Zusatzinformation%20NoSQL%20mit%20Volltextsuche.docx)

…

Das Graphmodel von Neo4j besteht aus Nodes, Relationships, Properties und Labels (vgl. Neo4j-Buch S. 74f) (siehe Abbildung 16).

Nodes beinhalten Informationen über Objekte. Es kann bei einer relationalen Datenbank mit einer eigenständigen Tabelle verglichen werden. Relationships sind Beziehungen zwischen den jeweiligen Nodes. Diese sind immer gerichtet, haben einen Start- und einen Endpunkt. Properties sind Eigenschaften. Diese können sowohl zu Nodes gehören, als auch zu Relationen. Im Falle von Nodes ist dies ähnlich einer relationalen Datenbank, in der in einer Tabelle mehrere Eigenschaften, z.B. Namen und gewisse Kennzahlen je Projekt enthalten sind. Zuletzt bieten Labels die Möglichkeit, Nodes zu kategorisieren. In Abbildung 16 ist dies durch verschiedene Farben dargestellt: „LKAB Sweden“ in rot ist ein Projekt, wohingegen alle grünen Kreise AddInformation sind. Die blauen gerichteten Pfeile sind Relationships.

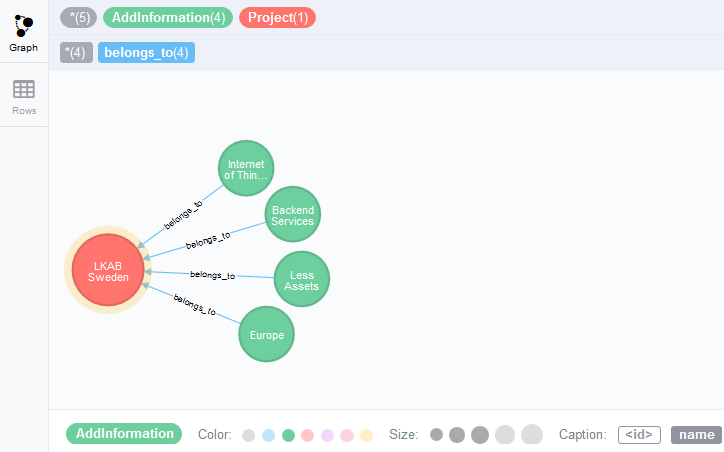
****

Abbildung Bestandteile von Neo4j

### Ergebnis

## Entwurfsphase des Wissensorganisationssystems

Im folgenden Kapitel wird der Entwurf der Datenbank mittels eines Domänenmodells vorgestellt. Die graphische Umsetzung auf Seite des Frontends wird durch Mock-Ups aufgezeigt. Die Systemarchitektur wird im darauffolgenden Kapitel der Realisierung des Prototypens erläutert.

### Domänenmodell der Projektdatenbank

In Abbildung 17 ist das logische Datenbankmodell der Projektdatenbank zu erkennen. Bei der Modellierung stand die Einfachheit der Struktur im Vordergrund. Diese ist notwendig, da eine NoSQL-Datenbank, wie es Neo4j ist, schemafrei ist und so offen für mögliche Erweiterungen sein muss.

Daher werden nur zwei Tabellen bzw. in Cypher-Sprache zwei Nodes angelegt: Project und AddInformation. Wird ein Project angelegt, erhält es automatisch eine ID. Ebenso ist es für die Anwendung der Graph Datenbank notwendig, dass ein sprechender Name angegeben wird. Alle weiteren Eigenschaften (Properties) sind optional. Der „type“ beschreibt, aus welchem Umfeld das Projekt stammt. Dies wird vor allem bei Projekten aus dem Buch angegeben. Bei Projekten aus der Excel-Tabelle sind andere Angaben vorhanden. Diese wurden ab dem Punkt „Project Dimensions“ mit einem zuvor angefügten Shortcut versehen, um besser auf die Informationen zugreifen zu können. So steht beispielsweise „ad\_g“ für den Unterpunkt „General“ in „Assets & Devices“, entsprechend der Tabellenstruktur.

AddInformation sind Tags, die via Open Calias (siehe Kapitel ….) generiert wurden. Beliebig viele können (mindestens) einem Projekt zugeordnet werden. Zudem ist es möglich, dass zwei oder mehr AddInformation die Beziehungen „is\_part\_of“ oder „same\_as“ enthalten. AddInformation können nicht auf sich selbst referenzieren.

Für beide unterschiedlichen Nodes gilt, dass ihre Informationen abgerufen werden können. Ebenso können ein Projekt oder eine AddInformation hinzugefügt, verändert und gelöscht werden.

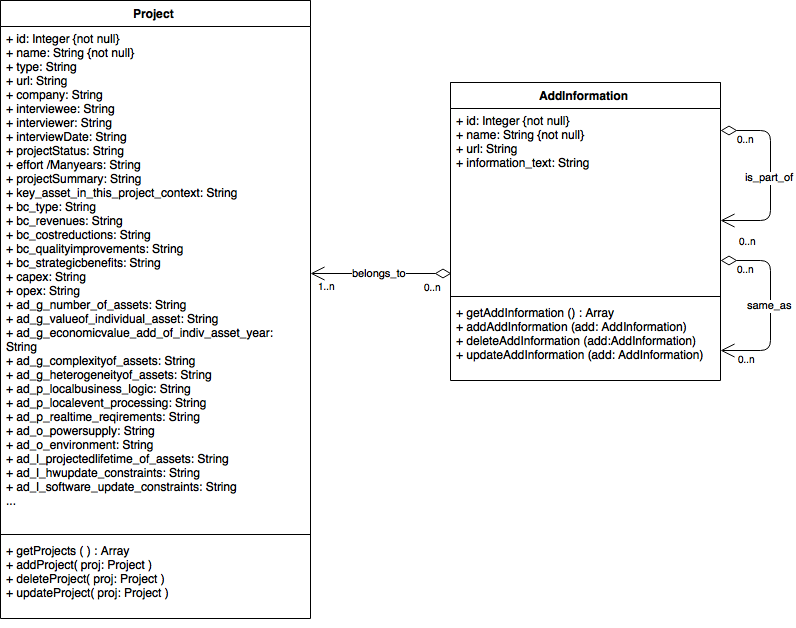


Abbildung Domänenmodell der Projektdatenbank

### Mock-Ups der Anwendung

Die Anwendung soll in einer eigenen Website realisiert werden. Das Design soll sich dabei dem der Buch-Website ähneln, um die Zusammengehörigkeit zum Kontext Ignite zu repräsentieren. Hierfür werden ähnliche Farben verwendet und das Ignite-Logo mit eingebunden. Wie in Abbildung 18 zu erkennen wird es eine Kopfleiste geben. Diese beinhaltet neben dem Logo eine Navigationsleiste mit den Seiten: Main, All Projects und Extended Search. Im Body werden dann die jeweiligen Inhalte dargestellt.

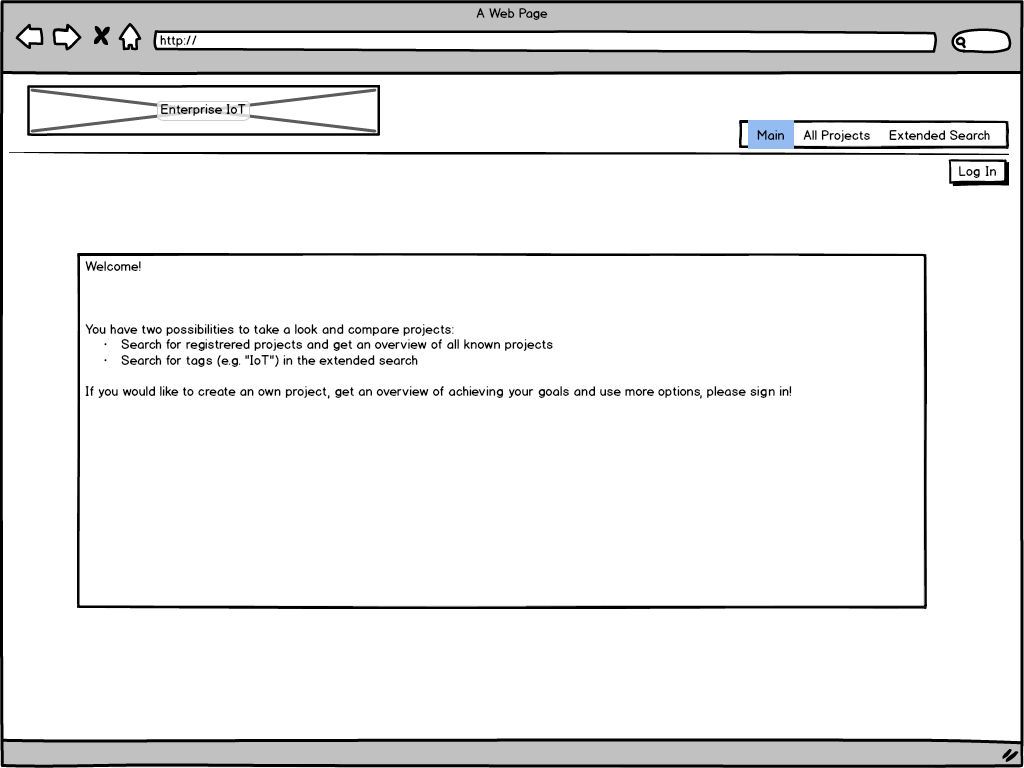


Abbildung - Startseite der Anwendung

Auf der Seite „All Projects“ werden zunächst alle Projekte untereinander angezeigt (siehe Abbildung 19 ). Es soll zudem möglich sein, nach einzelnen Projekten durch Eingabe des Namens suchen zu können.

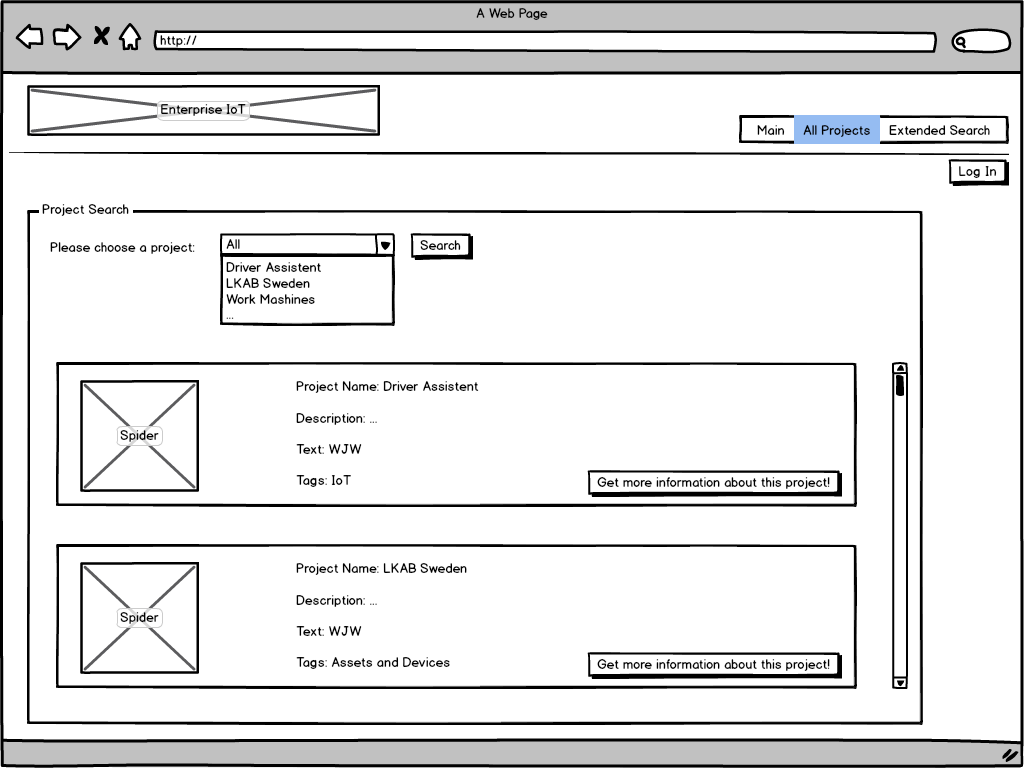


Abbildung – Suchseite

Für jedes Projekt besteht die Möglichkeit, sich detailliertere Informationen anzeigen zu lassen. Diese werden in einem Pop-Up-Fenster zusammengefasst (siehe Abbildung 20).

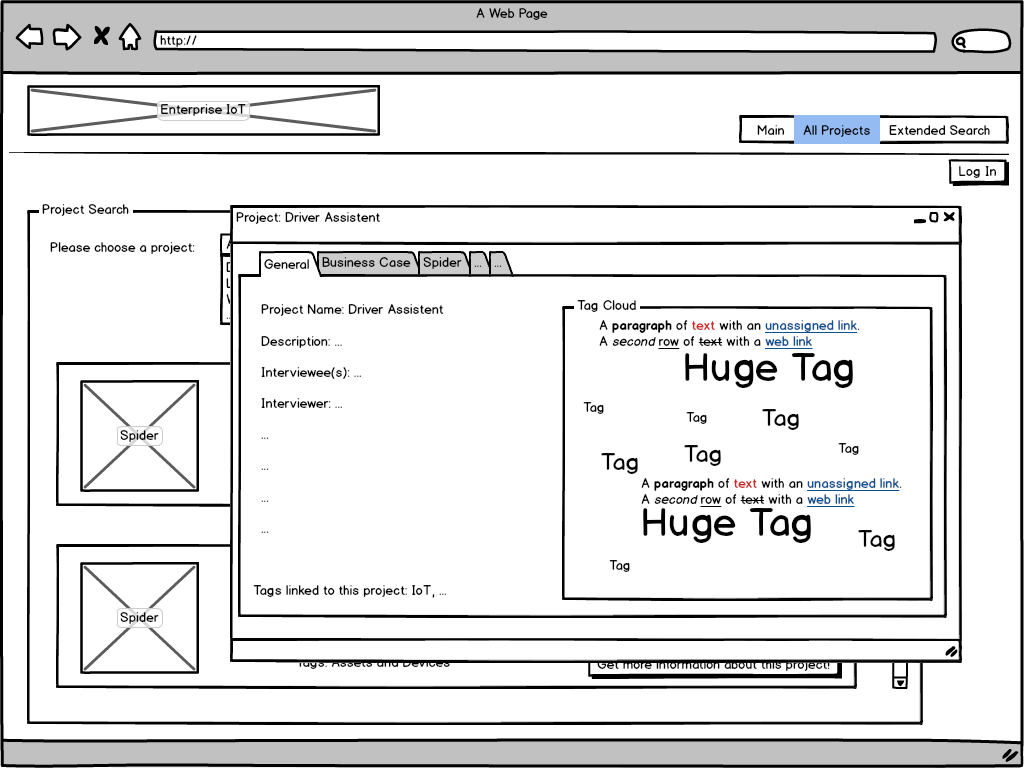


Abbildung - Detaillierte Projektinformationen

In der erweiterten Suche wird die Möglichkeit bestehen, sich die Umgebung von Projekten ausgeben zu lassen. Dafür kann entweder Freitext verwendet werden oder Checkboxen (siehe Abbildung 21)

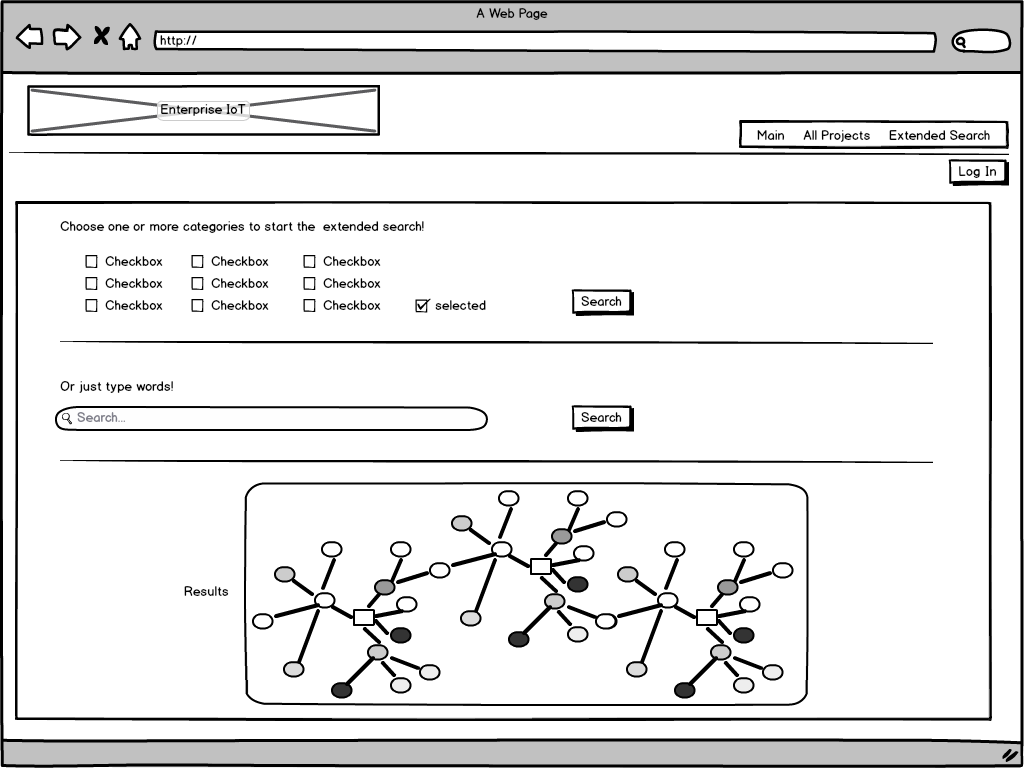


Abbildung - Erweiterte Suche

# Prototypische Umsetzung / Umsetzung der Anforderungen

Die prototypische Umsetzung fokussiert sich auf die Erstellung der Projektdatenbank im Backend. Das Frontend wird via AngularJS und HTML realisiert. Ein Zusammenspiel beider Komponenten erfolgt über eine REST-Schnittstelle, in der Cypher-Anfragen gestellt werden.

Die Realisierung dessen benötigt nachfolgende Schritte:

1. Backend: Neo4j-Datenbank aufsetzen & Daten implementieren
   1. Datenstruktur einbringen
   2. Grunddatensatz der vorhandenen Case-Studies des Excel-Sheets manuell implementieren
      1. Excel-Projekte einfügen
      2. Case Studies manuell implementieren ( = Voraussetzung zur Datensatzerweiterung)
   3. Datensatz erweitern
      1. Open Calias einsetzen
      2. Ergebnis in RDF speichern
      3. RDF durch inubit-Workflow verarbeiten
      4. CSV-Dateien generieren
      5. Daten implementieren
         1. CSV-Datei einlesen
         2. Generiertes Spreadsheet verwenden
   4. Abfragen zur erweiterten Suche entwickeln
2. Frontend entwickeln
   1. Grundstruktur erstellen
   2. Anbindung an Neo4j-Datenbank generieren
   3. Abfragen entwickeln
   4. Abfragen graphisch darstellen

Das Ergebnis wird aus einer Projektdatenbank und einer zugehörigen graphischen Darstellung bestehen. Die Datenbank wird Datensätze aus den Case Studies, sowohl aus der Excel-Tabelle als auch aus dem Buch enthalten. Diese werden mit generierten Tags angereichert und verbunden sein. Cypher-Abfragen werden mit diesem Datensatz verschiedene Anfragen abwickeln können und neue Erkenntnisse über die vorhandenen Projekte liefern können. Die graphische Darstellung wird sich auf ein Layout beschränken, indem es durch Verwendung von AngularJS und REST möglich sein wird, nach Projekten zu suchen, sich detaillierte Informationen ausgeben zu lassen, Projekte tabellarisch zu vergleichen sowie das Netz an Projekten und Zusatzinformationen optisch darzustellen. Der Prototyp wird auf keinem externen Server verfügbar, sondern nur über den localhost abrufbar sein.

## Softwarearchitektur

Abbildung Softwarearchitektur

## Entwicklung des Backends

Für die Entwicklung des Backends wird zunächst eine Neo4j-Datenbank aufgesetzt. Anschließend werden in dieser Daten implementiert.

### Anlegen der Datenbank & Grunddatensatz implementieren

Um Neo4j zu verwenden, ist eine Installation der Community Edition ausreichend. Ist dies erfolgt, werden die Serverumgebung sowie die benötigte Datenbank über das Startfenster ausgewählt und gestartet (siehe Abbildung 23) . Daraufhin kann Neo4j über den Port 7474 vom localhost aufgerufen.

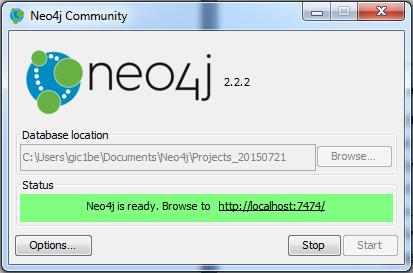


Abbildung Start von Neo4j

Ein erster Grunddatensatz der Projekte wird manuell implementiert. Hierfür werden einzelne Case-Studies aus dem Excelsheet via Cypher im Neo4j-Browser als Projects angelegt. Ebenso wird ein Projekt aus dem Buch hinzugefügt und erste AddInformations angelegt. Somit können bereits erste Relationen zwischen AddInformations und Projekten die Beziehungen verdeutlichen.

Die so generierte Grundstruktur dient als Voraussetzung für die nachfolgende Datensatzerweiterung.

### Datenimplementierung

Die Datenimplementierung erfolgt in zwei Schritten. So können neben der Anlegung von Daten gleichzeitig die Methoden zur Datenimplementierung evaluiert werden. Voraussetzung dessen ist der im Kapitel zuvor beschriebene Grunddatensatz.

#### Phase 1: Datenimplementierung mithilfe von „Learning Neo4j”

Im Buch “Learning Neo4j” werden mehrere Möglichkeiten gegeben, Daten in eine Neo4j-Datenbank zu implementieren.

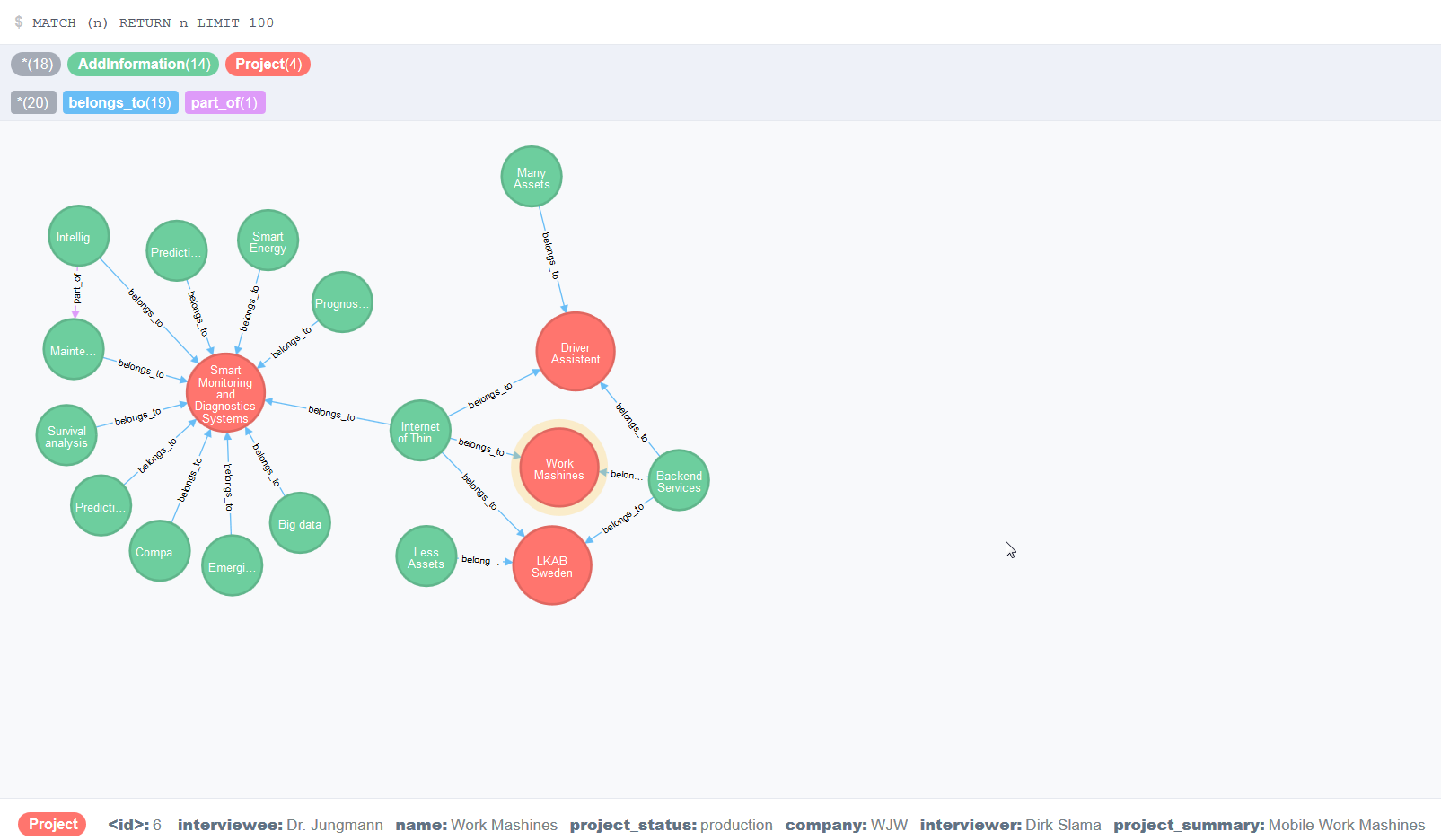


Abbildung – Ausschnitt des Datensatzes in der Projektdatenbank

In Abbildung 24 ist der Grunddatensatz zu erkennen. Dabei wurden drei Projekte des Excel-Sheets manuell implementiert. Ebenso wurde manuell das Projekt „Smart Monitoring and Diagnostics Systems at Major Power Plants“ samt erster AddInformation angelegt. Zu erkennen sind die einzelnen Beziehungen „belongs\_to“ zwischen Projekten und AddInformation. Ebenso wurde eine Beziehung „part\_of“ zwischen „Intelligent Maintenance Systems“ und „Maintenance“ definiert. Dieser Grunddatensatz wird nun mithilfe von Open Calias inubit erweitert.

In Open Calias wird je Case Study der vorhandene Text mit Metadaten angereichert (Die Funktionsweise wurde bereits in Kapitel … erläutert). Das Ergebnis wird als RDF konvertiert und nachfolgend in inubit angepasst.

Wie in Abbildung 25 ersichtlich, wird die lokal gespeicherte RDF-Datei via File Connector in inubit geladen. Die Datei wird validiert und anschließend konvertiert. Ergebnis dessen ist eine Auflistung aller benannten Tags aus Open Calias, welche nachfolgend als AddInformation genutzt werden. Das Ergebnis wird nach dem Splitter im oberen Abschnitt in das CSV-Format konvertiert und als CSV-Datei abgelegt. Im unteren Zweig werden je AddInformation jeweils Cypher-Queries erstellt: Zunächst wird eine AddInformation, falls noch nicht in der Datenbasis vorhanden, angelegt. Danach werden alle AddInformation mit dem Projekt verbunden.

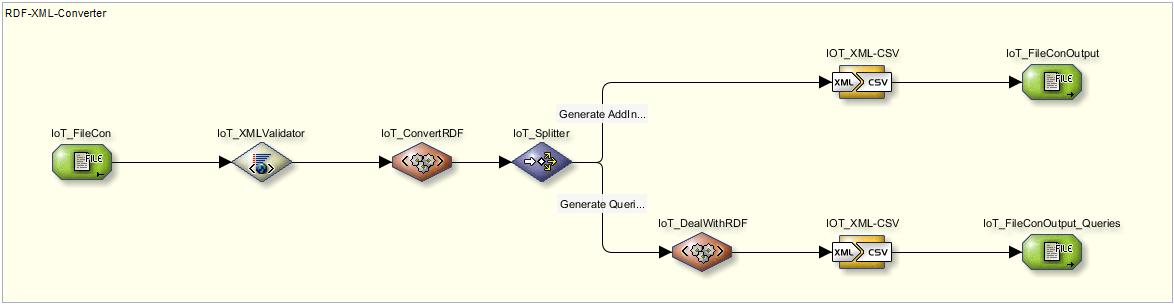


Abbildung inubit-Workflow zum generieren der CSV-Dateien

Das Ergebnis des oberen Zweiges kann via folgendem Befehl in die Datenbank implementiert werden:

|  |
| --- |
| LOAD CSV WITH HEADERS FROM "file:C:/Users/gic1be/Documents/Master-XML-Files/Output/Output\_14072015-111755.csv" AS row  MERGE(add:AddInformation {name: row.name})  RETURN add; |

Das Zwischenergebnis ist Abbildung 26 zu entnehmen.



Abbildung - Zwischenergebnis

Das Ergebnis des unteren Zweiges muss manuell kopiert und in den Neo4j-Browser eingefügt werden. Daraufhin hat sich der Datenbankinhalt wie in Abbildung 27 erweitert.

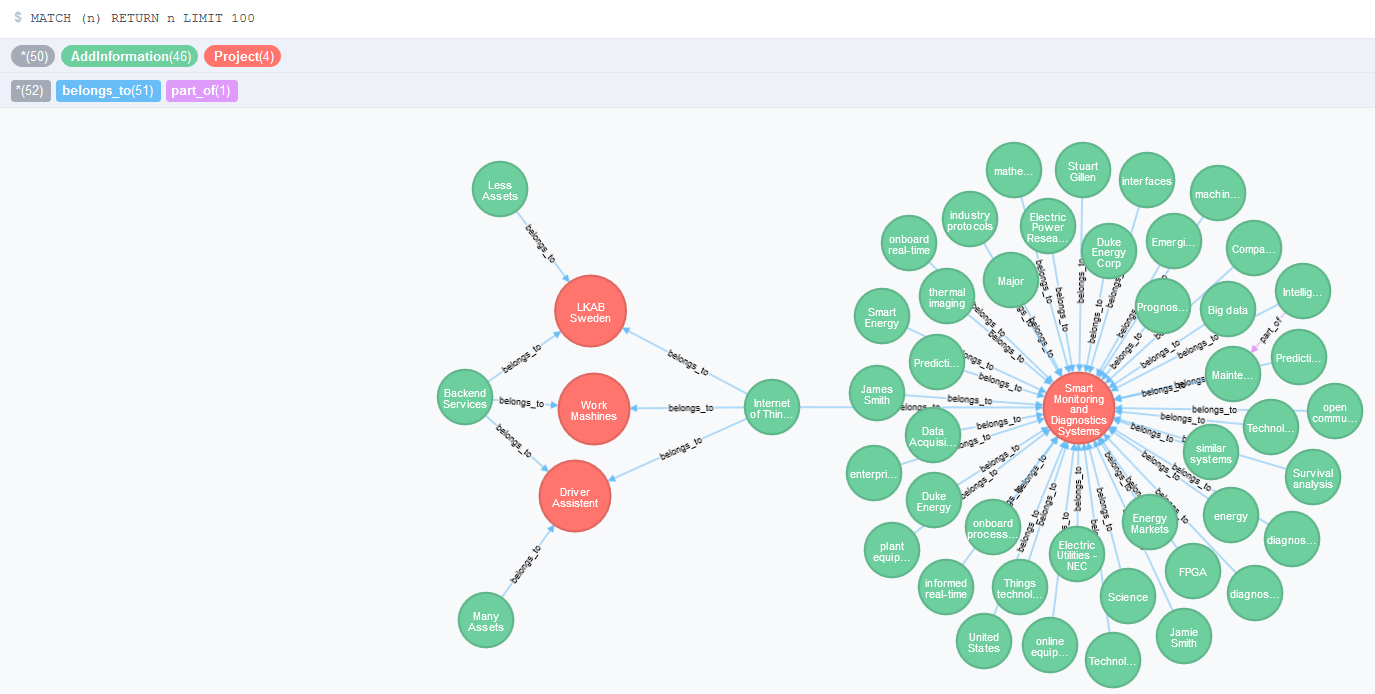


Abbildung Ergebnis CSV

Dieses Verfahren wird für alle Case Studies angewandt. Die Kombination aus automatisiertem Tagging via Open Calias, die Umwandlung des Ergebnisses in inubit sowie die Implementierung mithilfe des Neo4j-Browser auf technischer Seite – und nicht zuletzt der Einsatz des Expertenwissens, bilden zusammen den Grundstein zur Wissensmodellierung.

#### Phase 2

Um mehrere Projekte mit AddInformation anzureichern, wird eine Kombination aus generierten RDF-Dateien von Open Calias, einem inubit-Workflow und Cypher-Abfragen, welche „Load CSV“ verwenden, angewandt.

Im inubit-Workflow wird eine CSV-Datei generiert, die als Ergebnis Den Namen einer AddInformation enthält, sowie die zusätzlichen Informationen, dass es eine „property“ mit dem Inhalt „insert“ ist sowie ein value mit Wert 1. Der value-Wert soll später als Counter verwendet werden.

Die CSV-Datei wird via folgender Abfrage in Cypher in die Datenbank geladen:

|  |
| --- |
| LOAD CSV WITH HEADERS FROM "file:C:/Users/gic1be/Documents/Master-XML-Files/Output/2.csv" AS row  MATCH(add:AddInformation {name: row.name})  MATCH (add2:AddInformation)  WHERE NOT (add.name = add2.name)  MERGE(add:AddInformation {name: row.name, property: row.property, KindOfTag: row.KindOfTag})  RETURN add; |

Um die AddInformation einem spezifischen Projekt zuzuordnen, wird folgender Befehl angewandt:

|  |
| --- |
| MATCH (p:Project), (add:AddInformation) WHERE id(p)=8 AND HAS (add.property)  MERGE (add)-[:belongs\_to]->(p) |

Anschließend werden alle Eigenschaften mit Namen “Property” wieder entfernt:

|  |
| --- |
| MATCH (m {property:"insert"}) REMOVE m.property RETURN m |

Somit kann für jedes Projekt dessen spezifische AddInformation hinzugefügt werden.

|  |
| --- |
| LOAD CSV WITH HEADERS FROM "file:C:/Users/gic1be/Documents/Master-XML-Files/Output/Smart City.csv" AS row  MATCH(add:AddInformation {name: row.name})  MATCH (proj:Project {name:"Smart City"})  MATCH (add2:AddInformation)-[\*1..2]-thing  WHERE add.name=add2.name  MERGE (add)-[r:belongs\_to]-> (proj)  RETURN add2, proj |

Da mit diesem Befehl nur AddInformation in die Datenbank eingepflegt wurden, die noch nicht existieren, müssen bereits vorhandene AddInformation ebenfalls untersucht werden. Dabei werden alle AddInformation aus dem CSV mit der bereits vorhandenen Datenbasis verglichen. Falls

🡪 da wo es bereits eine Addinformation mit Namen Europe gibt.. verbinde das mit dem Projekt Smart City.

### Abfragemöglichkeiten mit Cypher

Cypher bietet, wie bereits in Kapitel … erläutert, nicht nur die Möglichkeit einfacher Abfragen. Das Potential steckt in Abfragen bezüglich verschiedenster Beziehungen und der Umgebungssuche. Diese werden im Prototypen aufgegriffen. So können nachfolgend Abfragen nach einzelnen oder mehreren Nodes direkt ausgeführt werden:

|  |  |  |
| --- | --- | --- |
| Rubrik | Frage | Cypher |
| Anzeigen | Zeige alle Projekte | MATCH p:Project RETURN p |
|  | Zeige alles an, was in direkter Umgebung zum Projekt liegt | MATCH (p:Project { name:"LKAB Sweden" })-[\*0..1]-(x)  RETURN x |
|  | Zeige alle AddInformation eines Projektes samt deren Beziehungen | MATCH (n:Project { name: 'LKAB Sweden' })-[r]-(a:AddInformation) RETURN n, r, a |
|  | Gibt es ein Projekt mit dem Zeichensatz x? | MATCH (p:Project)  WHERE p.name =~ 'Smart.\*'  RETURN p |
| Erstellen | Erstelle eine neue AddInformation mit dem Namen „Many Assets“ | CREATE (n:AddInformation{name:'Many Assets'}) RETURN n |
|  | Erstelle eine neue Beziehung zwischen der AddInformation „Many Assets“ und dem Projekt „Driver Assistent“, allerdings nur unter der Bedingung, dass diese Beziehung noch nicht existiert. | MATCH (add:AddInformation { name: 'Many Assets' }),(proj:Project {name: 'Driver Assistent'})  MERGE (add)-[r:belongs\_to]-> (proj)  RETURN r |
|  | Ändere eine Eigenschaft (http://Neo4j.com/docs/stable/query-set.html) | MATCH (p:Project { name: 'Driver Assistnt'})  SET Project.name='Driver Assistent'  RETURN p |
| Löschen | Lösche eine oder mehrere Projekte / AddInformation | MATCH (p:Project{name:“Project1“}) DELETE p |

Tabelle Einfache Cypher-Abfragen

Die Abfragen in Tabelle 5 können auch in ähnlicher Länge mit anderen Sprachen wie mySQL genutzt werden. Die Stärke von Cypher zeigt sich beispielsweise in den nachfolgenden Fragestellungen:

* Was ist in unmittelbarer Umgebung meines Projektes?
  + MATCH (proj:Project{name:"Driver Assistent"})-[\*1..2]-thing RETURN proj, thing
* Welche Projekte sind meinem ausgewählten ähnlich? (Stichwort Empfehlungen)
  + MATCH (proj:Project{name:"Driver Assistent"})-[\*1..2]-(thing:Project) RETURN proj, thing
  + In diesem Fall wird die nähere Umgebung des Projektes abgesucht. Ergebnis dessen sind nicht nur AddInformation, sondern auch die Relationen, die dort vorhanden sind (zum Beispiel, die Beziehung „belongs\_to“ zwischen AddInformation und Projekt)

|  |
| --- |
| MATCH  (tag:AddInformation)-[:belongs\_to]->(me:Project)  MATCH  (tag2:AddInformation)-[:belongs\_to]->(them:Project)  WHERE me.name = 'LKAB Sweden' AND tag.name=tag2.name AND NOT me=them  RETURN them.name AS Project\_name, count(\*) AS similar\_AddInformation  ORDER BY similar\_AddInformation DESC |

* + 🡪 In diesem Fall wird nicht die Umgebung abgesucht, sondern die gesamte Datenbasis. Ergebnis dessen sind nicht nur die Projektnamen, sondern auch eine Angabe über die Anzahl gemeinsamer Informationen. Sollen nicht nur die Projektnamen, sondern auch deren Properties angegeben werden, so ist bei der Abfrage lediglich ein „them“ zurückzugeben, ohne den Namen. Dadurch werde alle Informationen dessen zurück gegeben.
  + 🡪 Ein Problem dabei ist, dass die Relationen gerichtet sind. Daher musste die Abfrage geteilt werden. Ansonsten ginge es auch so wie hier:

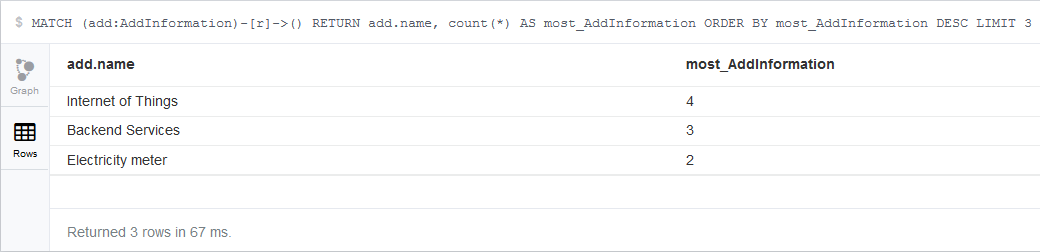
|  |
| --- |
| MATCH (me)-[:favorite]->(myFavorites)<-[:favorite]-(people)  WHERE me.name = 'Joe' AND NOT me=people  RETURN people.name AS name, count(\*) AS similar\_favs  ORDER BY similar\_favs DESC |

* Welche Projekte sind meinem ausgewählten ähnlich, haben aber auch die Eigenschaft X / ausgenommen Projekte mit der Eigenschaft X? (Stichwort Empfehlungen)

|  |
| --- |
| MATCH  (tag:AddInformation)-[:belongs\_to]->(me:Project)  MATCH  (tag2:AddInformation)-[:belongs\_to]->(them:Project)  WHERE me.name = 'LKAB Sweden' AND tag.name=tag2.name AND NOT me=them AND them.type="Smart Energy"  RETURN them.name AS Project\_name, count(\*) AS similar\_AddInformation  ORDER BY similar\_AddInformation DESC |

* Welche AddInformation wird von den meisten Projekten angesprochen?

|  |
| --- |
| MATCH (add:AddInformation)-[r]->()  RETURN add.name, count(\*) AS most\_AddInformation  ORDER BY most\_AddInformation DESC LIMIT 3 |

* + 
* Wie ist der kürzeste Weg von einem Projekt a (Connected Enterprise Solutions) zu einem Projekt b (LKAB Sweden)?
  + MATCH p=shortestPath( (proj1:Project {name:"Connected Enterprise Solutions"})-[\*]-(proj2:Project {name:"LKAB Sweden"})) RETURN p

## Entwicklung des Frontends

### Verwendete Komponenten (Angular JS)

Vor- und Nachteile nach Buch „AngularJS“ von O’Reilly (Seite 32f):

Plattformunabhängigkeit

(mobile und klassische Plattformen können verwendet werden)

Single Page Applications basieren auf Web-Technologien

Benutzerfreundlichkeit

User muss nicht auf das Neuladen der Seite warten

Durch JS Darstellung reichhaltiger Inhalte und Steuerelemente

Offlinefähigkeit

Siehe Buch, falls benötigt

Architektur

Passt zum Trend zu Service-basierten Systemen

Nachteile:

Nur wenn JavaScript zugelassen wurde

Unterschiedliche Browser erhöhen Komplexität und Testaufwand

Suchmaschinen können Seiten weniger gut indizieren

#### AngularJS

Populäres JavaScript-Framework zur Erstellung von Single Page Applications

Basiert auf Architekturmuster Model-View-Controller

# Evaluation

## Vorgehen

## Durchführung

## Auswertung

# Zusammenfassung

## Fazit

## Kritische Betrachtung

<http://www.moresophy.com/de/Search> bietet das schon an

BrainPro 8 (oder so) bietet das auch an

Kostet nur alles

## Nächste Schritte

Um den Prototypen als externe Anwendung nutzen zu können, bedarf es nachfolgender Schritte:

Die Datenbank muss extern erreichbar sein. Hierfür kann eine Cloudumgebung eingesetzt werden. Beispiele, in denen Neo4j Bereits als Lösung angewendet werden kann, sind GrapheneDB.com, Heroku, GraphAlchemist oder ElastX (vgl. Neo4j-Buch S. 65).

# Referenzen

# Begrifflichkeiten

Semantic/k

RDF

OWL

System – hier Synonym für Software

W3L

WM (Wissensmanagement)

WOS – Wissensorganisationssystem

Anhang

## Anhang A

1. http://wirtschaftslexikon.gabler.de/Definition/semantik.html [↑](#footnote-ref-1)
2. Quelle: Tim Berners-Lee, James Hendler, Ora Lassila: The Semantic Web,The Scientific American, 284(5), s. 34-43, 2001. [↑](#footnote-ref-2)
3. http://www.w3.org/2007/03/layerCake.png [↑](#footnote-ref-3)