Universität Rostock Traditio et Innovatio

WIRTSCHAFTSINFORMATIK
UNIVERSITÄT ROSTOCK

Konzept und prototypische Implementierung eines Recommendation Systems zur Unterstützung eines Lehr-Lern-Portals am Beispiel von myKOSMOS

eingereicht am 02.02.2015

von

Ingmar Fila | Steinfulgen 12 | 18069 Sievershagen

Matrikel-Nr.: 209204258

Gutachter:

Dipl.-Wirt. Inf. Dirk Stamer

Universität Rostock

Albert-Einstein-Str. 22

18059 Rostock

Zweitgutachter:

Dr. Meike Klettke

Universität Rostock

Albert-Einstein-Str. 22

18059 Rostock

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abstract

Empfehlungssysteme erfahren gerade in Lernportalen eine immer größere Beliebtheit. Sie können eingesetzt werden, um Lernenden passende Lehrmaterialien, Lernpartner bzw. Lerngruppen, Prüfungstermine oder andere wichtige Informationen anzubieten. Aus diesem Grund soll ein solches Empfehlungssystem im Rahmen eines Projektes der Universität Rostock realisiert werden, welches sich mit Erstellung eines kontextsensitiven Lehr-Lern-Portals befasst. Dazu wird im Rahmen dieser Arbeit ein Konzept eines Empfehlungssystems, welches dieses Portal als Ganzes unterstützen soll, entwickelt werden. Dazu wird das Portal sowie seine Aufgaben aus verschiedenen Blickwinkeln betrachtet, um ein möglichst umfassendes Bild für das Konzept zu bekommen. Zusätzlich wird ein Teil dieses Konzeptes prototypisch als *proof-of-concept* in diesem Portal umgesetzt. Diese Arbeit führt in die Grundlagen von Empfehlungssystemen, Informationslogistik sowie des verwendete Portals ein. Danach folgt das Konzept sowie die prototypische Umsetzung eines Teils dieses Konzeptes. Die Evaluation wird jeweils am Ende des entsprechenden Kapitels vorgenommen.

1 Einleitung

Die Motivation dieser Bachelor-Arbeit ist originär im Projekt *KOSMOS* der Universität Rostock zu finden. Im Rahmen dieses Projektes wird ein Konzept des sogenannten "Lebenslangen Lernens" umgesetzt und implementiert und so die mehr als zwanzigjährige Erfolgsgeschichte der *Alma Mater Rostochiensis* fortgesetzt. Die Universität Rostock möchte mit diesem Projekt die Bildungsbeteiligung stärken. Dadurch sollen neue Zielgruppen angesprochen werden, um diesen dann adäquate Studienmöglichkeiten auf universitärem Niveau bieten zu können. Dabei richtet sich dieses Angebot der Universität Rostock "[..] nicht nur an Menschen mit akademischer Erfahrung, sondern insbesondere an jene, die bisher keine, wenig oder länger zurück liegende Berührungspunkte mit einem Studium haben. [...]" [kos14]. Mit diesen neuen Studienformaten soll die Aufnahme eines Studiums und anderer universitärer Bildungsangebote in allen Lebensphasen ermöglicht werden.
Um das Projekt zu unterstützen, ist der Lehrstuhl Wirtschaftsinformatik unter der Leitung von Prof. Dr. Kurt Sandkuhl damit beauftragt worden, ein Portal für diese Zwecke zu entwickeln sowie zu implementieren. Das besondere Merkmal des Portals soll die Möglichkeit sein, den aktuellen Nutzerkontext zu erkennen und für die Anpassung der Portal-Inhalte zu verwenden. Für neue Zielgruppen und Studienformate an der Uni Rostock wird diese Kontextorientierung als Erleichterung im Studium angesehen. Um diese Nutzeranpassung zu ermöglichen und zu erleichtern wird im Rahmen dieser Bachelor-Arbeit ein Konzept für ein Recommendation System für das Portal *myKOSMOS* entwickelt. Dabei soll dieses Empfehlungssystem später für das gesamte Portal zuständig sein und die Nutzer in ihrem universitären Alltag zielgerichtet unterstützen.
Für die Erstellung des Konzeptes eines Empfehlungssystem zur Unterstützung der Nutzer des Portals *myKOSMOS* wird sich im Rahmen dieser Arbeit aus verschiedenen Richtungen genähert, um ein möglichst umfassendes Bild zu erhalten. Die erste Sichtweise wird daher die Betrachtung der Definition und des Zweckes des Portals sein, um dort schon Einsatzmöglichkeiten für ein Empfehlungssystem zu finden.

Daran anschließend findet eine Betrachtung aus Sicht der Prozessmodellierung statt, dabei wird der Ablauf eines Studentenjahres modelliert um sinnvolle Punkte zu finden an denen ein Empfehlungssystem ansetzen kann. Abschließend wird sich aus Richtung der technischen Seite genähert und die vorhandenen Datenstrukturen und Portlets auf Einsatzpunkte und -felder von Empfehlungssystemen untersucht. Diese Untersuchung findet sowohl auf der Ebene der vorhandenen und notwendigen Datenstrukturen statt, wie auch auf der Ebene der noch nicht vorhandenen aber notwendigen Datenstrukturen.
An dieser Stelle wird stichpunktartig erläutert, was nicht mehr Gegenstand dieser Bachelor-Arbeit sein wird. Dies umfasst:

- den Test des Prototyps unter realen Bedingungen
- die Effizienzoptimierung des Prototyps
- die optimale Fehlerbehandlung im Prototyp
- die vollständige Funktion des Prototyps
- die Konzeption einer Beobachtungskomponente

Unter realen Bedingungen ist der Betrieb des Portals und damit auch des Prototyps, gleichzeitig mit mehreren Nutzern oder Programmen, welche Nutzerverhalten simulieren, zu verstehen.

In dieser Bachelor-Arbeit werden in Kapitel 2 zunächst die Grundlagen erläutert, die für das Verstehen des Inhaltes dieser Arbeit notwendig sind. Im ersten Teil des Kapitels wird zunächst der Begriff des *Recommendation Systems* definiert. Es folgt einer kleiner Einschub zum Thema Gruppen, daran anschließend werden die Empfehlungssysteme mit Hilfe einer kurzen Übersicht klassifiziert und schematisch eingeordnet. Abschließend werden die drei wichtigsten Arten von Empfehlungssystemen näher vorgestellt. Im zweiten Teil des Kapitels wird auf die Thematik und Problematik der *Informationslogistik und -bereitstellung* genauer eingegangen. Im letzten Teil des Grundlagenkapitels wird auf das für die Konzeption und Implementierung verwendete Portal *Liferay* eingegangen. Ebenso wird eine kurze Übersicht über zusätzliche, durch andere Arbeiten erstellte Portlets im Portal *myKOSMOS* geben.

Im darauf folgenden Kapitel 3 wird die für die Erarbeitung des Konzeptes notwendige Forschungsmethodik und das Vorgehen erläutert. Dies umfasst die mehrseitige Untersuchung des Portals unter verschieden Gesichtspunkte sowie die daraus folgenden Konsequenzen.

In Kapitel 4 wird das Konzept des Empfehlungssystems näher vorgestellt. Dabei wird sich aus den im Kapitel 3 beschriebenen Richtungen genähert, um ein umfassendes Bild zu erhalten. Daran anschließend werden wünschenswerte Erweiterungen des Portals vorgestellt, die zukünftig das Empfehlungssystem unterstützen sollen. Ebenfalls Bestandteil des Konzeptes ist der Aspekt des Datenschutzes. Der darauf folgende Abschnitt hat die Präsentation der Empfehlung im Portal als Inhalt. Im vorletzten Abschnitt wird die notwendig der Überprüfung gegebener Empfehlungen erläutert und diskutiert. Der letzte Abschnitt ist schließlich der Bewertung des Konzeptes gewidmet.

Mit der prototypischen Implementierung eines Teiles des Konzeptes aus Kapitel 4 beschäftigt sich das Kapitel 5. Der erste Abschnitt 5.1 enthält die für die Implementierung verwendete Software. Der zweite Abschnitt beschäftigt sich mit der eigentlichen Implementierung eines Teils des Konzeptes aus 4. Die Bewertung der Implementierung wird wie beim Konzept ebenfalls im letzten Abschnitt vorgenommen.

Abschließend wird alles Relevante in Kapitel 6 unter bestimmten Gesichtspunkten im ersten Abschnitt entsprechend ausgewertet und noch einmal kurz erklärend zusammengefasst. Schlussendlich wird im letzten Abschnitt, welcher diese Literaturarbeit beschließt, noch ein Blick in die Zukunft des Portals geworfen.

2 Grundlagen

Dieses Kapitel wird die Einführung in die Grundlagen enthalten, welche notwendig sind um das in Kapitel 4 vorgestellte Konzept und die darauf folgende Implementierung eines Teils dieses Konzeptes in Kapitel 5 umfassend zu verstehen.

2.1 Recommendation Systeme

Dieser Abschnitt führt in das Thema der *Recommendation Systeme* oder auch Empfehlungssysteme ein. Dazu wird zuerst eine Definition für Empfehlungssysteme erarbeitet, es folgt ein kleiner Einschub zum Thema der Gruppen, anschließend werden Empfehlungssysteme klassifiziert. Abschließend für dieses Kapitel werden dann die drei populärsten Ansätze (inhaltsbasiert, kollaborativ und hybrid) für Empfehlungssysteme genauer beschrieben.

2.1.1 Definition

In der Literatur zum Thema *Recommendation Systeme* lassen sich eine Vielzahl an Definitionen finden. Im Grunde ist ein Empfehlungssystem ein System, welches Nutzern Empfehlungen ausspricht [HK07]. Diese Definition ist jedoch sehr trivial, weshalb im nachfolgenden Definitionen betrachtet werden, welche einen formelleren und präziseren Charakter besitzen.

Eine solche Definition lässt sich in der Forschungsarbeit von [AT05] finden. Ihrer Ansicht nach ist die Aufgabe eines Empfehlungssystems die Lösung des Optimierungsproblems:

$$s'_c = max_{s \in S}(u(c, s))$$

für einen gegebenen Nutzer $c \in C$.

Mathematisch betrachtet: Einzelnen Personen aus einer Menge von Nutzern C werden Elemente aus einer Ressourcenmenge S empfohlen. Eine Bewertungsfunktion $u : C \times S \rightarrow \mathbb{R}$ bildet jede Nutzer-/Elementkombination auf eine geordnete Menge, wie beispielsweise die reellen Zahlen, ab. Mit diesem Maß wird versucht, die Nützlichkeit eines Elementes für einen bestimmten Benutzer abzubilden. Sie definieren ein Recommendation System als die Maximierung eines Nutzwertes in Abhängigkeit von einem Nutzer und einer vorgegebenen Ressourcenmenge.

Eine zweite ähnliche Definition wird von [Kla09] aufgestellt, welche ebenfalls ein Empfehlungssystem über eine Nutzwertmaximierung definiert:

$$s'_c = max(Nutzwert(c, K, s))$$

mit $K = (P, M, S)$.
Als Addition zur ersten Definition wird noch eine Variable "Kontext" K eingeführt, welche sich aus einem "Benutzerprofil" P, der "Entitätsmenge" M und der "Situation" S zusammensetzt. Das "Profil" P kann aus expliziten Informationen (Geschlecht, Alter, Interessensgebiete, usw.), als auch aus impliziten Informationen (Besuchshäufigkeit einer Website, gelesene Texte, gekaufte Produkte, etc.) bestehen.
Die "Entitätsmenge" M zählt zum Kontext, da sich auch alleine aus ihrer Änderung (Neuzugang von Entitäten) und ohne Änderung des Profils eine Empfehlung ergeben kann.
Die "Situation" S enthält Parameter der realen Welt wie Datum, Uhrzeit, Geoinformationen oder Informationen über das verwendete Endgerät des Benutzers.

Es ist jedoch zu beachten, dass beide Definitionen von einem Nutzer ausgehen. Offensichtlich existieren jedoch auch Empfehlungssysteme, welche Empfehlungen an mehrere Personen aussprechen. Der Nutzwert muss also nicht nur für einen Nutzer, sondern für eine ganze Gruppe maximiert werden.
In Anlehnung an die erste Definition lässt sich so ebenfalls eine Definition für Empfehlungssysteme für Gruppen finden:

$$s'_c = max_{s \in S}(u(c_1, ..., c_n, s))$$

Diese Definition und jene von [Kla09] werden im weiteren Verlauf dieser Arbeit genutzt.

2.1.2 Gruppen

An dieser Stelle soll ein kleiner Einschub in die Thematik der *Gruppen* einführen. Dies ist für das weitere Verständnis der Recommendation Systeme notwendig, da diese Einzelpersonen sowie auch Gruppen Empfehlungen aussprechen können. Für das Konzept selber steht die einzelne Person im Vordergrund, allerdings lassen sich andere, womöglich bessere Empfehlungen generieren, wenn sich diese Person einer Gruppe zuordnen lässt.

Die Mathematik definiert eine Gruppe als eine Menge von Elementen, die über eine zweistellige innere Verknüpfung verbunden sind, durch die jedem geordneten Paar von Elementen eindeutig ein Element dieser Menge als Resultat zugeordnet wird, wenn diese Verknüpfung assoziativ ist und es ein neutrales Element gibt, sowie zu jedem Element (s)ein Inverses existiert [Lau11].

Eine weitere Definition lautet: Eine Gruppe ist eine Ansammlung einer Mehrzahl von Personen (mindestens zwei [PO09]), die über einen längeren Zeitraum in direktem Kontakt steht. Jedes Mitglied muss sich zudem der anderen Mitglieder bewusst sein, zwischen den Mitgliedern muss eine Interaktion möglich sein. Die Rollen differenzieren sich innerhalb der Gruppe durch den direkten Kontakt aus. Zudem entwickeln sich in einer Gruppe gemeinsame Normen, wodurch sich ein "Wir-Gefühl" (auch *Kohäsion* genannt) ausbildet. Außerdem ist die Gruppengröße nicht nach oben begrenzt (nach [Ner08]).

Damit grenzt sich die Gruppe in der Definition von einer Organisation ab. Organisationen sind zeitlich relativ stabile, gegenüber der Umwelt offene, aus Individuen und Gruppen zusammengesetzte, zielgerichtet handelnde und strukturierte Systeme [Ner08].

Im weiteren Verlauf dieser Arbeit wird die soziologische Gruppendefinition von [Ner08] genutzt.

2.1.3 Klassifizierung

In Abschnitt 2.1.1 wurde der Begriff des Empfehlungssystems definiert. In diesem Abschnitt wird es darum gehen, diesen Begriff in weitere Kategorien einzuteilen, um eine Einordnung von Empfehlungssystemen sowohl für Gruppen als auch für Einzelpersonen vorzunehmen. Hierzu werden die verschiedenen Arten von Empfehlungssystemen aufgezeigt werden.

[HK07] teilen Empfehlungssysteme grundsätzlich in personalisierte und nicht-personalisierte Empfehlungssysteme ein. Letztere sind zum Beispiel *Information-Retrieval-Systeme* wie Suchmaschinen oder die digitalisierte Suche in einem Bibliothekskatalog. Der entscheidende Unterschied zu personalisierten Empfehlungssystemen ist, dass alle Nutzer bei gleicher Eingabe in das System dieselbe Empfehlung bekommen; Präferenzen oder sonstige Eigenschaften des Nutzers finden keine Beachtung. Es ist also gleich, ob die Eingabe von einer einzelnen Person stammt oder aber als Konsens von einer Gruppe eingegeben wird, wodurch eine weitere Differenzierung an dieser Stelle unerheblich ist.

Personalisierte Empfehlungssysteme können auf verschiedene Weisen eingeteilt werden. In der Literatur ist es üblich, sie nach dem dahinterliegenden Ansatz zur Empfehlungsbestimmung zu unterscheiden. Die verbreitetsten Ansätze sind der inhaltsorientierte sowie der kooperative (auch kollaborative) Ansatz; weniger geläufig sind der demografische und der wissensorientierte Ansatz. Weiterhin existieren auch Kombinationen, sog. hybride Ansätze, aus allen vier genannten Ansätzen. Die Ansätze können (wenn auch angepasst) sowohl auf Gruppen als auch auf Einzelpersonen angewandt werden, weshalb jeder Ansatz dahingehend nochmal unterschieden wird.

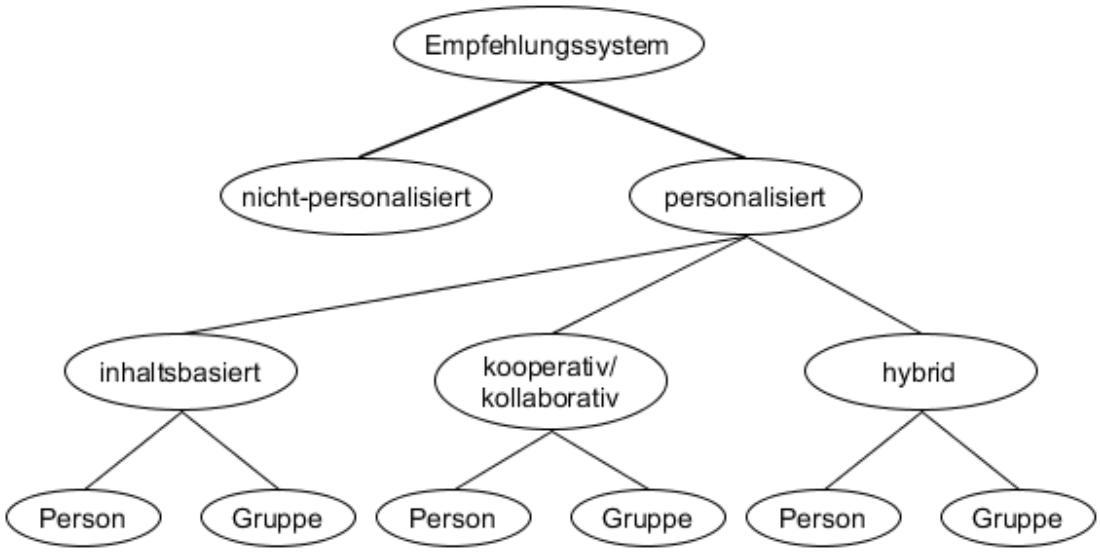

Abbildung 2.1: Klassifizierung, eigene Darstellung nach [HK07]

[MC07] entwickelten ein Rahmenwerk zur Klassifizierung von Empfehlungssystemen, welches auch andere Möglichkeiten der Einteilung bietet. So könnte man beispielsweise Empfehlungssysteme auch nach der Art der zu erfüllenden Aufgabe, der Entscheidungsproblematik, der dahinterliegenden Architektur oder nach der Art der Auslösung (Push- oder Pullsystem) kategorisieren. In der Literatur ist es jedoch unüblich, Empfehlungssysteme nach dem dahinterliegenden Ansatz zu unterscheiden, weshalb auf eine weitergehende Klassifizierung innerhalb dieser Arbeit verzichtet wird.

In den folgenden Abschnitten werden die verbreitetsten Ansätze aus Abbildung 2.1 umfassender dargestellt.

2.1.4 Inhaltsbasierte Empfehlungssysteme

Dieses Verfahren, welches auch als *content-based filtering* bzw. *feature-based filtering* bekannt ist [Mar12], basiert im Wesentlichen auf den Eigenschaften der Empfehlungselemente [Kla09]. Die Eigenschaften der Elemente werden auf ihre Ähnlichkeit mit den Eigenschaften der Präferenzen des jeweiligen Nutzers verglichen, wobei das Verhalten anderer Nutzer überhaupt keine Rolle spielt.

Im Rahmen der Eigenschaftsanalyse werden die Eigenschaften bestimmt, welche dann zur Selektion nützlicher Elemente für eine Empfehlung verwendet werden. Dabei wird die Nützlichkeit oft mit der Ähnlichkeit zu Elementen im aktuellen Kontext, der Situation oder im Profil der Nutzers gleichgesetzt [Mar12].

Um eine Ähnlichkeit zu bestimmen, kann das Korrelationsmaß nach Pearson angewendet werden. Der Pearson-Korrelationskoeffizient ist eine Ähnlichkeitsfunktion Sim für zwei Vektoren v und w mit $v = (v_1, v_2, ..., v_n)$ und $w = (w_1, w_2, ..., w_n)$ gilt:

$$Sim_{cos\theta}(v, w) = \frac{\sum_{i=1,...,n}((v_{i-} - \overline{v}) * (w_{i-} - \overline{w}))}{\sqrt{\sum_{i=1,...,n}(v_{i-} - \overline{v})^2 * \sum_{i=1,...,n}(w_{i-} - \overline{w})^2}}$$

Dabei stehen $\overline{v}$ bzw. $\overline{w}$ für den Durchschnitt der Komponenten von v bzw. w [Kla09].

Inhaltsbasierte Empfehlungssysteme geben Nutzern Empfehlungen, die eine inhaltliche Gemeinsamkeit mit einem von ihm genutzten oder gut bewerteten Element haben. Dafür erstellt das System für jeden Benutzer ein eigenes Nutzerprofil, das Informationen über Präferenzen in Form von codierten Schlüsselbegriffen enthält [Fel10]

2.1.5 Kooperative Empfehlungssysteme

Bei kooperativen bzw. kollaborativen Empfehlungssysteme beruhen die Empfehlungen, im Gegensatz zu inhaltsbasierten Empfehlungssystemen, nicht auf den Präferenzen des Nutzers selbst. Die Empfehlung wird beim kooperativen Ansatz aus den Präferenzen anderer Nutzern abgeleitet, welche üblicherweise in einer $M \times N$ Matrix gespeichert werden (siehe Abbildung 2.2). Die erste Dimension steht dabei für Elemente, die Zweite für die Nutzer.

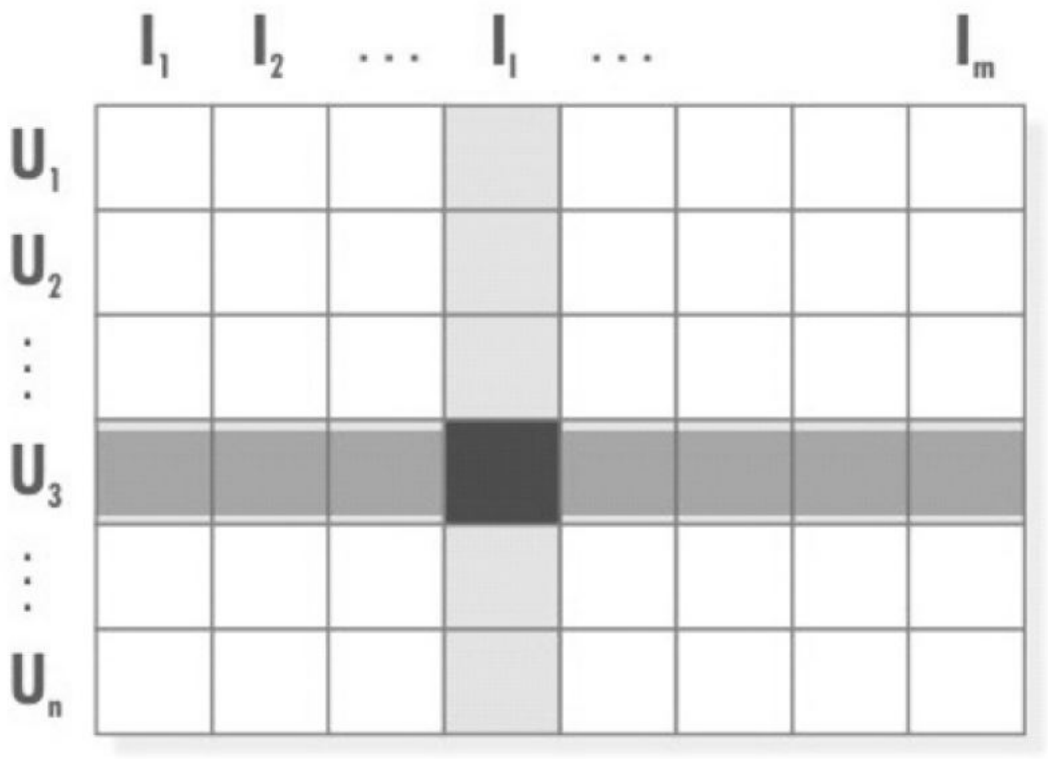

Abbildung 2.2: Matrix zur Speicherung von Präferenzen, Quelle: [Kla09]

Für die Berechnung einer Empfehlung mit diesem Ansatz existieren eine Vielzahl an verschiedenen Algorithmen, welche sich jedoch in zwei Hauptgruppen kategorisieren lassen. Auf der einen Seite sind das speicherbasierte (auch *memorybased*/*heuristic-based*) und auf der anderen musterbasierte (auch *modelbased*) Algorithmen.

Speicherbasierte Algorithmen arbeiten mit Heuristiken, die auf Grundlage der gesamten Menge an Bewertungen aller Nutzer eine Vorhersage über die Bewertung r eines Produkts s des Nutzers c tätigen. Diese Vorhersagen werden meist über eine Aggregation der Bewertungen anderer Nutzer berechnet. Letztgenannte sind gewöhnlicherweise diejenigen Nutzer, welche dem Nutzer c am ähnlichsten sind. Formal lässt sich dies wie folgt darstellen:

$$r_{c,s} = aggr_{c' \in \hat{C}} r_{c',s}$$

Die Aggregation ist hierbei nur ein Platzhalter für verschiedene Methoden, welche die Bewertungen zusammenzufassen. Nachfolgend werden zwei Möglichkeiten einer Aggregation aufgezeigt.

Eine sehr einfache Möglichkeit ist, den Durchschnitt der Bewertungen zu bilden:

$$r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s}$$

Diese Technik ist jedoch nur brauchbar, wenn ein Element von allen anderen Nutzern ausschließlich positiv oder negativ bewertet wurde. Neutrale Bewertungen führen zu schlechten Ergebnissen bei Empfehlungen, weshalb diese Technik in der Praxis eher unüblich ist.

Am Häufigsten wird eine Technik angewandt, bei der die Summe eine Gewichtung erhält:

$$r_{c,s} = k \sum_{c' \in \hat{C}} sim(c, c') \times r_{c',s}$$

wobei $k = 1/\sum_{c' \in \hat{C}} |sim(c, c')|$.
Hierbei wird $sim(c, c')$ benutzt um den Grad der Ähnlichkeit zwischen zwei Nutzern zu messen. Je größer dieser Grad ist, umso mehr Gewicht erhält die Bewertung bei der Berechnung der Vorhersage, was deren Korrektheit wahrscheinlicher macht. Solche Ähnlichkeiten werden häufig über ein Korrelationsmaß (z.b. nach Pearson) bestimmt.
Bei modellbasierten Verfahren wird nicht die vollständige Matrix mit den gespeicherten Nutzerpräferenzen verwendet. Stattdessen wird ein vereinfachtes Modell wie zum Beispiel eine “Benutzer-Cluster“ Matrix verwendet [Kla09].

Zu den bekanntesten modellbasierten Ansätzen gehören *Bayes'sche Netze* sowie *Clusterbildung*, bei welchem Nutzer mit ähnlichen Vorlieben gruppiert werden. Auf Basis dieser Gruppen werden dann Empfehlungen für den Nutzer berechnet, indem die Präferenzwerte aller Benutzer eines Clusters, in denen sich ein Nutzer befindet, zu einem Durchschnittswert verdichtet werden.

Dieser Wert kann potenziell auch übergreifend auf Basis von Präferenzwerten der Benutzer anderer Cluster, in denen sich der aktive Benutzer ebenfalls befindet, ermittelt werde, wobei die Präferenzwerte in diesem Fall mit dem Grad der Cluster-Zugehörigkeit gewichtet werden [Kla09].

Der kollaborative Ansatz benötigt eine gewisse Anzahl von Bewertungen als Ausgangsbasis um korrekt und präzise arbeiten zu können. Sind zu wenige Bewertungen (Präferenzen) vorhanden können im *Worst-Case-Scenario* keine oder nur unpräzise und damit qualitativ minderwertige Empfehlungen berechnet werden. Dieses Problem ist als sog. *Kaltstartproblem* oder auch *New-User-Problem* bekannt. [Fel10, Ahn08]

2.1.6 Hybride Empfehlungssysteme

Werden inhaltsbasierte und kooperative Empfehlungssysteme zusammen verwendet, so spricht man von einem hybriden Empfehlungssystem. Diese Systeme sind recht modern und deshalb noch wenig verbreitet. Sie versuchen durch das Kombinieren von inhaltsbasierten und kooperativen Empfehlungssystemen, die Probleme und Einschränkungen beider Systeme abzumildern oder gänzlich zu vermeiden. [Mar12]
Bei dieser Art von Empfehlungssystemen werden deshalb kooperative Empfehlungssysteme um die inhaltsbasierten Interessenprofile der Nutzer erweitert. Durch diese Profile wird es hybriden Empfehlungssystemen möglich, neben den Kundeninteressen auch Gemeinsamkeiten zwischen Produkteigenschaften zu erkennen, anstatt, wie bei kooperativen Systemen üblich, nur Ähnlichkeiten zwischen Kunden zu registrieren. [Kla09]

Neben der Erweiterung der Empfehlungssysteme durch Profile, können die Empfehlungen auch getrennt berechnet werden und über verschiedene Strategien kombiniert werden. Je nach Situation könnte es sinnvoll sein, dass das Ergebnis einer der beiden oben beschriebenen Ansätze priorisiert wird. Die Ergebnisse können ebenfalls einfach gemischt werden, sodass Empfehlungen zu jeder Zeit sowohl vom einen als auch vom anderen Ansatz stammen können. Eine dritte Strategie ist das Zusammenfassen der beiden Empfehlungsberechnungen zu einem einzigen Ergebnis für Empfehlungen, wobei es zusätzlich möglich ist, Gewichtungsfaktoren zu verwenden. [Bur02]

Folglich können Empfehlungen sowohl auf Basis der Präferenzen eines einzelnen Benutzers (inhaltsbasiert), als auch auf der Basis abgegebener Bewertungen anderer Nutzer mit ähnlichen Präferenzen (kooperativ/kollaborativ) gegeben werden. Die Genauigkeit von hybriden Empfehlungssystemen liegt dadurch signifikant ($p < 0,001$) höher als bei inhaltsbasierten oder kooperativen Filtersystemen, wie erste Untersuchungen zeigten [Mar12]. Zudem steigt durch hybride Empfehlungssysteme die Qualität der Empfehlungen [Fel10].

Trotz dieser Vorteile unterliegen auch hybride Empfehlungssysteme dem von den kollaborativen Empfehlungssystemen bekannten *New-User-Problem* [Fel10].

2.2 Informationslogistik und -bereitstellung

Dieser Abschnitt soll der Einführung in die Thematik und Problematik der Informationslogistik und der bedarfsorientierten Informationsbereitstellung dienen. Zuerst wird der Begriff der *Informationslogistik* näher erläutert, anschließend auf den Aspekt der *bedarfsorientierten Informationsbreitstellung* eingegangen.

2.2.1 Informationslogistik

Die Informationslogistik wird immer wichtiger, da die heutige globale Vernetzung durch Informations- und Kommunikationstechnologien aller Art eine Überflutung mit Informationen ausgelöst hat. Dies stellt sich vorrangig in der Form dar, dass durch diese Vernetzung Informationen von jedem Standort auf der Welt zu jedem Zeitpunkt verfügbar sind und auch sein müssen. Diese Ubiquität von Informationen konfrontiert den Nutzer mit einer schier unübersichtlich Menge an Informationen, was es selbigem schwer macht, wichtige Informationen zu identifizieren, zu verarbeiten und dann richtig anzuwenden. Und genau an diesem Punkt setzt die Informationslogistik an, um der Informationsüberflutung einen Ansatz zur Eindämmung entgegenzustellen.

Die Informationslogistik erfuhr im Laufe der letzten Jahre viel Aufmerksamkeit von Seiten der Forschung und ebenso von Seiten der Anwender. Damit gut nachvollzogen werden kann, worauf sich die Definition der Informationslogistik nach [MAGM13] bezieht, soll an dieser Stelle der Begriff der *Information* an sich erklärt werden.

Eine Information wird im Allgemeinen als “eine [...] gegenwarts- und praxisbezogene Mitteilung über Dinge, die [...] im Augenblick wichtig zu wissen sind“ [K+05] beschrieben, speziell in der Literatur wird dabei zum großen Teil die kürzere Definition von Information von [Wit59] genutzt, die Information als “zweckorientiertes Wissen“ definiert. Zusätzlich wird an dieser Stelle der Begriff der Information vom Begriff des Wissens, der Nachrichten und der Daten abgegrenzt. “Wissen ist die Gesamtheit der Wahrnehmungen, Erfahrungen und Kenntnisse eines Menschen oder einer Gruppe von Menschen über sich und die Umwelt beziehungsweise einen Teilbereich daraus.“ [LWS08]. Nachrichten sind alle Mitteilungen zwischen zwei Menschen, “deren Struktur durch ein gemeinsames System von Symbolen bestimmt ist“ [LWS08] Diese können mithilfe von Daten übertragen werden (siehe Abbildung 2.3).

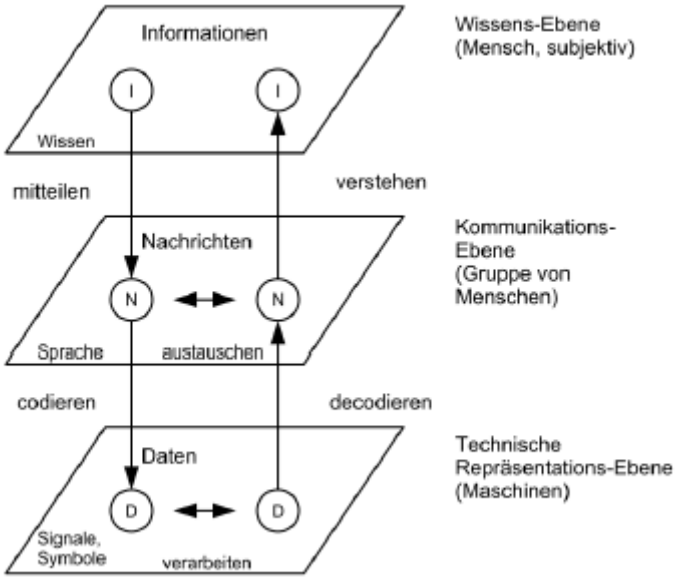

Abbildung 2.3: Hierarchischer Zusammenhang zwischen Informationen, Nachrichten und Daten [HBVBK94]

Nachdem der Begriff der Information definiert und von anderen Begriffen aus diesem Kontext abgegrenzt wurde, soll nun die Definition der Informationslogistik erfolgen.

[MAGM13] definieren und beschreiben das Ziel der Informationslogistik nun in wenigen einfachen aber sehr präzisen Worten: Die **effektive und effiziente** Versorgung (Bereitstellung) der

- richtigen Informationen
- im richtigen Format, der richtigen Granularität und der richtigen Qualität,
- zur richtigen Zeit,
- am richtigen Ort
- für den richtigen Empfänger.

Diese Definition erscheint auf den ersten Blick wie viele andere Definitionen zum Thema Informationsversorgung. Hierbei ist es wichtig zu beachten, das im Gegensatz zur bedarfsorientierten Informationsbereitstellung die Informationslogistik an sich noch einen weiteren Teil, nämlich den Teil der Informationsbeschaffung, mit einschließt.

2.2.2 Bedarfsorientierte Informationsbereitstellung

Die bedarfsorientierte Informationsbereitstellung leitet sich aus dem vorherigen Abschnitt "Informationslogistik" her. Die bedarfsorientierte Informationsbereitstellung lässt sich nach dem vorherigen Abschnitt 2.2.1 als ein Teil der Informationslogistik betrachten, da die Informationslogistik im allgemeinen die Beschaffung und die anschließende Verteilung der Daten mit einschließt. Dabei muss beachtet werden, dass sich dieser Abschnitt mehr auf bedarfsorientierte Informationsbereitstellung im Portal *myKOSMOS* konzentrieren wird und dadurch weniger allgemein gehalten wird.
[MAGM13] unterscheiden in ihrer Arbeit zwischen 10 verschiedene Arten (Gruppen) von informationslogistischen Ansätzen. Diese sind, in der Reihenfolge des Auftretens in der Arbeit:

- C1: Strategie und Management
- **C2: Nutzerorientierte Informationslogistik**
- **C3: Prozessorientierte Informationslogistik**
- **C4: Informationslogistik Prozess**

- **C5: Agentenbasierte Informationslogistik**
- C6: e-Maintenance
- C7: Wissens-Management
- C8: Frühwarn-Systeme
- C9: Kollaboration
- C10: Supply Chain Management

Die in dieser Aufzählung hervorgehobenen Arten sind die für diese Arbeit relevanten Arten der Informationslogistik. Relevant bedeutet in diesem Zusammenhang, dass die genauere Betrachtung dieser Ansätze zur Informationsbereitstellung als Teil der Informationslogistik für das weitere Verständnis dieser Arbeit notwendig ist. Dies umfasst das genaue Verständnis, wie Empfehlungen von Recommendation Systemen unter den in Abschnitt 2.2.1 genannten Aspekten an einen Nutzer weitergegeben werden können, um eben jene Probleme zu vermeiden, welche die Informationslogistik und die bedarfsorientierte Informationsbereitstellung zu lösen versuchen. Die hervorgehobenen Arten werden nun näher erklärt.

Den in **C2** zusammengefassten Arten der Informationslogistik ist gemein, dass bei dieser Gruppe die Informationsversorgung von Nutzern im Vordergrund steht. Dabei geht es einmal um die Herausforderungen die damit einhergehen, aber auch um Lösungen für diese Herausforderungen. Nach [DLP03] kann so ein Teil Informationslogistik als just-in-time Bereitstellung von Informationen betrachtet werden. In der Arbeit von [BG10] wird darauf eingegangen, das der Erfolg von Informationsversorgung auf der erfolgreichen Annahme durch den Nutzer und einer leistungsstarken Präsentation basiert. Ebenso wird der Kontextorientiertheit der Informationslogistik eine große Rolle zugemessen, von [LLSS06] wird beispielsweise ein Modell eines solchen kontextbasierten Informationslogistik-Modells ausführlich vorgestellt.

In **C3** wurden Ansätze für Prozessorientierte Informationslogistik zusammengefasst. Diese Ansätze haben als Aufgabe, prozessrelevante Informationen, wie beispielsweise Arbeitsanweisungen, Best Practices und ähnliches, für wissensintensive Geschäftsprozesse passend zur Verfügung zu stellen, so dass Entscheidungsträger und Arbeiter, die viel Wissen benötigen, ihre Aufgaben in der möglichen Art und Weise verrichten können [MMR12]. Genauer gesagt, prozessorientierte Informationslogistik ermöglicht prozessorientierte und kontextorientierte Bereitstellung von relevanten Informationen für wissensintensive Arbeiten. Um dies zu erreichen wurde ein Netzwerk semantischer Informationen genutzt, welches Prozessobjekte, Informationsobjekte und deren Beziehungen untereinander zusammenfasst [MAGM13].

In der Gruppe **C4** wurden alle Informationslogistischen Prozesse zusammengefasst, welche vorrangig von der *Jönköping Business School* in Schweden bearbeitet werden [MAGM13]. Diese grenzen sich von der Gruppe **C3** dadurch ab, als dass sie von [HA07] als ein Ansatz (genauer als ein Prozess) eingeführt wurden, um eine Eingabe, z.B. eine Projektbeschreibung, ein "Lesson-learned"-Dokument, etc., in eine Art von Ausgabe, beispielsweise ein "Best-Pratices"Dokument, zu überführen. Das Ziel hinter diesem Prozess ist es verteilte (fragmentierte) Informationen in eine für einen Empfänger nutzbare Information zusammenzufassen. [MAGM13]

Den Abschluss sollen die in **C5** zusammengefassten Ansätze bilden. Diese Ansätze sind der agentenbasierten Informationslogistik zuzurechnen. In diesem Kontext ist ein Agent eine Software, welche für einen Nutzer benötigte Informationen sucht. [BWZV07] präsentiert einen beispielhaften Ansatz eines agentenbasierten Ansatz zum Überwachen und Koordinieren von Prozessen. [MAGM13]

Wie nach diesen Definitionen ersichtlich ist, haben diverse Arten von informationslogistischen Ansätzen einen Nutzer oder eine Gruppe von Nutzern als Mittelpunkt in ihrem Konzept. Da das Portal *myKOSMOS* ebenfalls Nutzer bzw. Nutzergruppen als Mitglieder hat, welche in Zukunft durch ein Empfehlungssystem bedient werden sollen, so sind die oben genannten Ansätze die wichtigsten für diese Aufgabe, in Verbindung mit der Definition der Informationslogistik aus Abschnitt 2.2.1.

2.3 Portal Liferay

In diesem Abschnitt wird die von *myKOSMOS* verwendete Portal-Software *Liferay* erklärt. Dabei wird zuerst eine kleine Übersicht über die Thematik der Portale an sich gegeben, daran anschließend wird dann die verwendete Portal-Software *Liferay* erklärt.

Ein Portal oder Portalsystem ermöglicht den Zugriff auf gebündelte Informationen und Anwendungen von diversen Datenquellen. Portale lassen sich als Webanwendungen realisieren und für verschiedene Zwecke einsetzen, beispielsweise sind Community-Portale zur Verwaltung der Aktivitäten einer Internetcommunity oder Wissensportale in Unternehmen, mittels derer Wissen gesammelt und verteilt werden kann [Leh14]. Desweiteren gibt es Portale im Bereich des E-Learning, die als Lehr- und Lernplattformen dienen [Str12]. Der Aufbau eines Portals ist in Abbildung 2.4 dargestellt. Die Inhalte eines Portals werden in Form von Portlets bereitgestellt, dies sind Elemente in einem Portal, die über bestimmte Funktionalitäten (Suche, Anzeigen von Dokumenten, etc.) verfügen. Durch die Verwendung von Standards wie der *Java Specification Request* ([jsr15]), auch als *Java Portlet Specification* bekannt, ist die Verwendung eines Portlets in verschiedenen Portalsystemen möglich. Entwicklern ermöglicht die *Java Portlet Specification*, Portlets zu entwickeln, die unabhängig von einem bestimmten Portal zur Nutzung bereit gestellt werden können. Administratoren können dadurch aus einer Menge von Portlets gezielt die Portlets auswählen, die den Anforderungen des eigenen Portals entsprechen [Leh14, Str12].

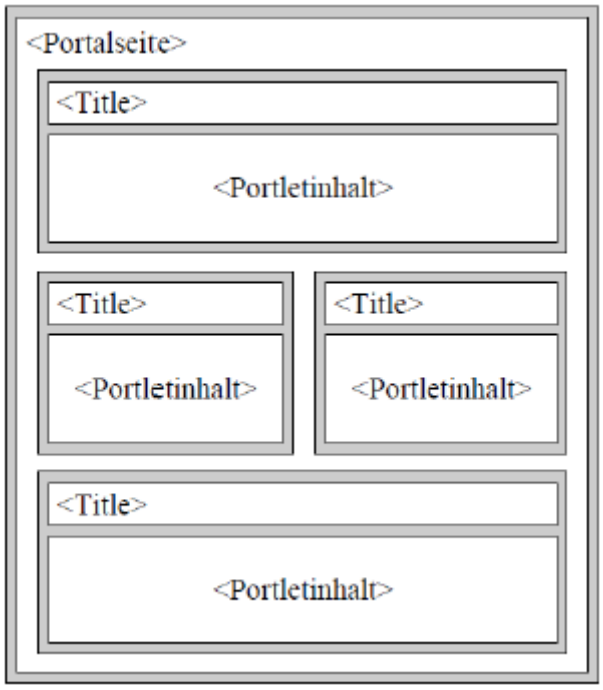

Abbildung 2.4: Layout einer Portalseite [Leh14]

[Str12] beschreibt verschiedene Anforderungen an ein Portalsystem. Eine zentrale Forderung ist, dass das Portal einen zentralen Login, einen sogenannten *Single-Sign-On*, bieten muss. Dadurch wird dem Benutzer ein mehrmaliges Authentifizieren vor der Benutzung von verschiedenen Portlets im Sinne der Benutzerfreundlichkeit erspart. Durch die Verwendung einer einheitlichen sowie benutzerfreundlichen Oberfläche wird den Nutzern eine einfache Bedienung ermöglicht, da selbige sich dann schneller zurecht finden können. Den Benutzern muss es außerdem möglich sein, mit nur einem Klick zwischen den Portlets zu wechseln.
Als weitere wichtige Eigenschaft wird von [Leh14] die “Personalisierung“ hervorgehoben. Dadurch wird z.B. in einem Benutzerprofil festgelegt, welche Portalelemente in welcher Anordnung in dem Portal angezeigt werden.

Das Portal Liferay selbst dient, wie jedes andere Portal, als populäre Beispiele seien hier *iGoogle* sowie *NetVibes* genannt, dazu, bestimmte Funktionalitäten (Anwendungen) und/oder Zugänge zu Informationsquellen unter einer kohärenten Oberfläche zu bündeln. Dies geschieht unter anderem auch, um vor dem Nutzer zu verbergen, dass verschiedene Anwendungen dahinter liegen. Wie bereits im Einleitungskapitel 1 beschrieben, soll das besondere Merkmal des Portals *myKOSMOS* die Möglichkeit sein, den aktuellen Nutzerkontext zu erkennen und dadurch für die Anpassung der Portal-Inhalte zu sorgen, dies umfasst die Portlets und deren Inhalt.
Das Portal *myKOSMOS* der Universität Rostock nutzt zum Dar- und Erstellen eine existierende Software-Plattform für Portale. Dabei handelt es sich dabei um die Software *Liferay*. Liferay selber ist eine lizenzkostenfreie Open Source Software und wird unter der "GNU Lesser Generel Public Licencse" verwendet. Diese Software ist ein auf der Programmiersprache *Java* basierendes Portalentwicklungstool, welches (Stand: November 2014) große und relevante Java-Anwendungsserver, Betriebssysteme (bis auf MacOS) und Datenbanken (kein MSAccess) unterstützt. Die genauen Spezifikationen von *Liferay* können auf der Seite [lif14] eingesehen werden.
In Liferay selbst können durch die Bündelung von Funktionalitäten in JavaServerPages sogenannte Portlets erstellt werden. Diese Erfüllen im Portal die Funktion von Unterprogrammen und bieten zusätzliche Funktionalitäten zu den bereits vorhandenen Portlets.

Nachdem nun die Thematik der Portale und *Liferay* erläutert wurden, ist ersichtlich, dass die Erkenntnisse für Empfehlungssysteme und die bedarfsorientierte Informationsbereitstellung für die Konzeption und Umsetzung eines Empfehlungssystems in einem Portal unumgänglich sind. Einmal konzipiert kann ein Empfehlungssystem auf diese Art und Weise auf verschiedene beliebige Standards übertragen werden, was zu einer hohen Flexibilität in der Anwendung in Portalen führt.
Für die Umsetzung des in dieser Arbeit konzipierten Empfehlungssystems sind der Standard *JSR* sowie die Portalsoftware *Liferay* notwendig.

2.4 Portal myKOSMOS

In diesem Abschnitt wird speziell auf die im Rahmen des Projektes “KOSMOS“ bereits für das Portal *myKOSMOS* entwickelten Portlets eingegangen werden. Dies umfasst eine kurze Beschreibung des Nutzens und der Funktionen der einzelnen Portlets.

Request (Wegtam-Suche)

Die *Request* oder auch *Wegtam-Suche* benannte Suchmaschine wird vom Portal *myKOSMOS* als Alternative zu gängigen Suchmaschinen im Internet eingesetzt. Die Integration der Suchmaschine wurde im Rahmen der studentischen Arbeit am Portal der Universität Rostock konzipiert und implementiert. Wegtam ist eine Rostocker Firma, die mit ihrer Suchmaschine dem Nutzer die Möglichkeit geben will, verfügbare Ressourcen schneller abzurufen und die wirklich relevanten Inhalte schneller zu finden. Dies geschieht dadurch, dass Wegtam auf der einen Seite in seiner Funktion als Meta-Suchmaschine (wie z.B. IxQuick), also als eine Suchmaschine (Suchagent), die Suchanfragen an andere Suchmaschinen gleichzeitig stellen und die gesammelten Ergebnisse aufbereiten, bereits vor der Suche die Möglichkeit bietet, durch Suchprofile Suchmaschinen oder Kategorien auszuwählen, um so seine Suche von vornherein gezielter ausrichten zu können. Die Ergebnisse der Suchanfrage können anschließend durch Filter (Stichwort, Website, Suchmaschine, Dokument, etc.) von Nutzer dynamisch eingeschränkt werden [weg15e]. Die Suchmaschine von Wegtam selbst, welche im Jahr 2012 neu gestaltet und allen Nutzern weltweit zugänglich gemacht wurde, ist zu finden unter [weg15c]. Das besondere an dem Wegtam-Suchagenten ist, das jede Version bestimmten Grundsätzen unterliegt [weg15d]. Diese verhindern das Abspeichern von Nutzerdaten, sodass nie Rückschlüsse auf die Erkenntnisse eines bestimmten Nutzers gezogen werden können. Ebenso werden keine Daten über Suchanfragen oder sonstige personenbezogene Informationen an etwaige Dritte weitergegeben. Als letzter Punkt werden, um Nutzer bei jeder Suchanfrage durch den Wegtam-Suchagenten nach besten Wissen und Gewissen zu schützen, Suchanfragen anonymisiert, so dass keine Suchanfrage einem bestimmten Nutzer zugeordnet werden kann. Das Portlet im Portal *myKOSMOS* greift auf eine speziell für die Arbeit im Portal auf akademische Nutzung angepasste Version [weg15a] der Wegtam-Suche zu. Diese unterliegt den selben Auflagen wie die normale Wegtam-Suche auch. Das *Request*-Portlet sendet die Suchanfragen von der Portal-Oberfläche direkt an den Wegtam-Suchagenten der Universität Rostock [weg15a], und verfügt bereits über 45 eingebundene Suchmaschinen [weg15b].

Support2-portlet

Das *Support2-Portlet*-Portlet wurde im Rahmen der Master-Arbeit von [Ack14] entworfen und implementiert. Es handelt sich bei diesem Portlet um eine Empfehlungskomponente, welche auf Basis von Daten über Nutzer und Regeln einem Nutzer Portlets empfiehlt.
Dieses Portlet hat als Basis zwei Komponenten: eine Boebachtungskomponente und eine Empfehlungskomponente. Die Beobachtungskomponente überwacht Aktivitäten im Portal, wie beispielsweise die Eingabe eines Suchbegriffes, Maus-Klicks oder die Veränderung (hinzufügen, entfernen, vergrößern, verkleinern) von Portlets. Diese Aktivitäten werden in einer von [Ack14] entworfenen Datenstruktur abgespeichert. Die Empfehlungskomponente wertet nun diese Daten nach entsprechenden vorgegebenen Regeln aus und generiert über diesen Weg Empfehlungen für den jeweiligen Nutzer, in diesem Falle werden Empfehlungen für Portlets ausgesprochen. Bisher geschieht dies allerdings nicht selbstständig, sondern der Empfehlungsprozess muss vom Nutzer selber angestoßen werden. Da dies nicht sehr komfortabel ist, muss hier entsprechend weitergearbeitet werden.

Skype Chat & IM

Das *Skype Chat & IM*-Portlet wurde im Rahmen des Projektes "Port2Port " von [DVWG14] als Erweiterung des Portals *myKOSMOS* in Liferay entworfen und implementiert.
Bei diesem Portlet handelt es sich um eine Komponente, die im Portal vorhandene Gruppen durch Umsetzung von Grundfunktionen von Skype, z.B. Video-Chats und Verfügbarkeitsanzeigen unterstützen soll. Um die Gruppenunterstützung effizient zu gestalten, wurden die Funktionen in einem Portlet gebündelt. Im Rahmen der Entwicklung des Skype-Portlets von [DVWG14] wurde ein bereits bestehendes Portlet von [por15] gefunden und um diverse Funktionen erweitert. In dem bestehenden Portlet von [por15] war es möglich, Personen zu Gruppen hinzufügen und diese dann zu speichern, über Skype anzurufen und anzuschreiben. Jedoch waren diese Gruppen in dem Original-Portlet privat und konnten so nur von den Mitgliedern der jeweiligen Gruppe gesehen werden, Mitglieder die außerhalb der Gruppe standen waren nicht in der Lage diese zu sehen. Um diesen Umstand zu verbessern, wurden von [DVWG14] die sogenannten "Shared Groups" eingeführt, bei denen jedes Mitglied einer Gruppe mitgeteilt bekommt, wenn es zu einer Gruppe hinzugefügt wurde, was dementsprechend dazu führt, dass das betreffende neue Mitglied nun auch direkt über Skype Kontakt aufnehmen kann.

Zusätzlich wurde die Möglichkeit implementiert, dass in den Suchergebnissen gefundene Personen nun direkt angerufen werden können. Ebenso ist es nun möglich, den Online-Status dieser Personen zu überprüfen, allerdings können Personen ihren Online-Status mittels der Einstellungen von Skype verbergen, was es demzufolge unmöglich macht, den Status der betroffenen Personen von außerhalb zu überprüfen.
Um den Erfolg weiterführender Arbeiten am "Skype"-Portlet zu gewährleisten, wurde ein weiteres Portlet von [DVWG14] konzipiert und implementiert, welches "Events", die vom eigentlichen "Skype"-Portlet bei verschiedene Aktionen gesendet werden, empfangen und darstellen kann.

My Google Drive Quick Links

Das *My Google Drive Quick Links*-Portlet wurde im Rahmen der Master-Arbeit von [Wei14] entworfen und implementiert.
Dieses Portlet verbindet die Nutzung des Portals mit der Nutzung des Cloud-Dienstes *Google Drive* direkt im Portal. Das Portlet verfügt dabei über fast alle Funktionen, die der Cloud-Dienst *GoogleDrive* anbietet, also den schnellen Up- und Download von Dateien aus der Cloud. Es stellt über eine Verknüpfung zu existierenden *google Drive*-Accounts deren Inhalte zur Verfügung, dabei ist es zum Zeitpunkt dieser Arbeit momentan nicht möglich, schnell zwischen zwei verschieden Accounts zu wechseln oder sie zu verknüpfen.

StudIP-Anbindung

Das StudIP-Portlet wird zum Zeitpunkt der Bearbeitung dieser Arbeit im Rahmen der studentischen Arbeiten am Portal *myKOSMOS* geschaffen, in der Arbeit von [Spe15] wird das Portlet anschließend vom Entwickler selber bewertet, aber nicht analysiert.
Der Zweck dieses Portlets lässt sich kurz beschreiben: Ankündigungen und Änderungen aus dem StudIP der Universität Rostock sollen im Portal zugänglich gemacht werden, um so dem Nutzer eine zentralisierte Informationsmöglichkeit zu geben. Bisher ist es nicht möglich beispielsweise Ankündigungen aus dem StudIP auf das Portal zu übertragen. Dieses Problem soll nun durch die Einführung dieses Portlets gelöst werden. Damit wird es dem zukünftigen Nutzer des Portals leichter gemacht, mit wenig Aufwand an wichtige Informationen zu gelangen. Dies dient der Nutzerfreundlichkeit, in dem ein zentraler Anlaufpunkt für Informationen geschaffen wird.

3 Forschungsmethodik und Vorgehen

In diesem Kapitel wird die Methodik und das Vorgehen erläutert, mit dem das Konzept in Kapitel 4 erstellt wird. Dieses Konzept soll ein Recommendation System für das Portal *myKOSMOS* erstellen, welches universell im gesamten Portal einsetzbar ist.

Zum Erstellen des Konzeptes müssen die für ein Recommendation System wichtigen Komponenten **user** und **items** identifiziert werden. Um diese Komponenten zu identifizieren, muss das Portal von allen Seiten betrachtet und untersucht werden.
Im Rahmen der Untersuchung dieses Portals wird sich diesem von drei Seiten genähert, diese Seiten umfassen:

- den Einsatzzweck (die Definition) des Portals,
- die prozessorientierte Sichtweise,
- und die technische Sichtweise (diese umfasst weitere Unterteilungen).

Als Erstes wird der Einsatzzweck des Portals untersucht werden. Dazu wird die Definition des Projektes *KOSMOS* von der Website der Universität Rostock genutzt. Daran anschließend werden aus dieser Übersicht alle Einsatzzwecke, die ein Recommendation System benötigen, durch ein solches unterstützt werden können oder ein Empfehlungssystem durch z.B. Daten unterstützen können, herausgearbeitet.

an den ersten Schritt schließt sich teilweise der zweite Schritt bei der Erarbeitung des Konzeptes an. Dieser Teil des Konzeptes wird auf Basis der Prozessorientierung und -modellierung entstehen.
Dazu werden mehrere beispielhafte Studenten modelliert, die in diesem Portal arbeiten und ein (oder mehrere) Semester durchlaufen. Dieses Durchlaufen wird als Prozess modelliert, auch auf Basis der Einsatzzwecke des Portals.

Anschließend werden alle Punkte in diesem Prozess identifiziert, an denen ein Recommendation System sinnvoll und hilfreich ist. Ebenso müssen alle dafür notwendigen Daten (**items**) identifiziert werden, die ein Recommendation System benötigt, um eine sinnvolle Empfehlung ausgeben zu können.

Der letzte Schritt bei der Konzeption soll den technischen Aspekt umfassen. Da es auch bei der technischen Sichtweise mehrere Möglichkeiten der Betrachtung gibt, die sich auch teilweise stark überschneiden, sollen diese entsprechend vorgestellt werden. Eine Sichtweise umfasst die Betrachtung der bereits vorhandenen Portlets, die andere Sichtweise umfasst die Datenstrukturen des Portals.
Im Bereich der Portlets wird zuerst auf die Systemfunktionalitäten eingegangen, die *Liferay* mit sich bringt. Dies umfasst die bereits vorhandenen Portlets sowie Datenstrukturen in der *Liferay* Grundinstallation. Daraus können bereits einige für ein Recommendation System wichtigen Daten **user** und **items** gewonnen werden.
Eine weitere Betrachtungsweise stellt die Untersuchung der durch die Arbeit an *myKOSMOS* entstandenen Portlets und Datenstrukturen dar. Diese werden ebenfalls auf für Recommendation Systeme wichtige bzw. sinnvolle Datenstrukturen untersucht.
Ein weiterer Schritt bei der Konzeption im technischen Bereich läuft parallel zu den beiden anderen Arbeitsvorgängen ab. Dieser Schritt umfasst die Untersuchung der Datenstrukturen auf Vollständigkeit. Das bedeutet, dass die Datenstrukturen dahingehend überprüft werden, ob sie bereits alle notwendige Attribute für die Arbeit eines Recommendation Systems aufweisen. Ist dies nicht der Fall, so wird ein Vorschlag für eventuelle Zusätze an der Datenstruktur und damit am Portal selber unterbreitet werden. Diese Vorschläge werden entweder im selben Abschnitt an den entsprechenden Stellen gegeben oder bei einem entsprechend großen Umfang in Abschnitt 4.4 gesammelt, um dort genauer behandelt zu werden.

Ebenfalls nicht bei der Konzeption vergessen werden darf der Aspekt des Datenschutzes. Ein Empfehlungssystem arbeitet immer zum einem gewissen Teil mit sensiblen Daten, diese müssen ausreichend geschützt werden. Wie dieser Schutz aussehen kann, wird in Abschnitt 4.5 ausführlich behandelt.

Ein weiteren wichtigen Aspekt bei der Konzeption stellt das Problem der Präsentation da. Wie können Empfehlungen am besten in einer Portalumgebung präsentiert und platziert werden? Sollen Empfehlung nahtlos integriert oder (deutlich) als solche gekennzeichnet werden? Welche Möglichkeiten bieten, für diese Arbeit und das Portal *myKOSMOS*, konkret *Java* bzw. *Liferay* bei der Umsetzung der Präsentation? Gibt es schon Portale oder andere Websites, die eine gute Präsentation von Empfehlungen anbieten oder gibt es Portale, wo dies nicht gut gehandhabt wird?
All dies sind wichtige Fragen, die in Abschnitt 4.6 des Konzeptes beantwortet werden sollen.

Abschließend für dieses Konzept soll die Problematik der Auswertung von Empfehlungen diskutiert werden. Es muss die Möglichkeit geben, Empfehlungen auf ihre Qualität, also auf ihre Präzision hin zu überprüfen. Wie könnte man dies bewerkstelligen? Welche Probleme gibt es dabei? Welche Verfahren bieten sich an und welche bereits bestehenden Möglichkeiten gibt es?
Diese Fragen werden ebenfalls im Konzept berücksichtigt und finden in Abschnitt 4.7 ihren Platz.

4 Konzept

Dieses Kapitel soll der näheren Erläuterung des Konzeptes für die Anbindung eines *Recommendation Systems* an das Portal dienen. Wie bereits in Kapitel 3 dargestellt, wird sich der erste Abschnitt aus der Sichtweise der Definition des Portals nähern. Daran anschließend folgen die Sichten aus der Prozessorientierung und der technischen Ebene. Ebenso werden in diesem Kapitel wünschenswerte Erweiterungen, der Aspekt des Datenschutzes, die Präsentation der Empfehlung sowie deren Qualitätsbewertung behandelt. Abschließend wird das Konzept als Ganzes unter verschiedenen Gesichtspunkten bewertet.
An dieser Stelle sei erwähnt, das sich das Konzept nur mit der Empfehlungskomponente des gesamten Systems beschäftigt. Da Nutzerdaten vorrangig über Beobachtung der Selben gewonnen werden müssen, sind diese Beobachtungen in eine Datenstruktur zu überführen, an der das Empfehlungssystem anknüpfen kann. Diese Beobachtungskomponente muss im Rahmen einer weiteren Arbeit am Portal geschaffen werden. In diesem Konzept werden allerdings rudimentär Anknüpfungspunkte für eine solche Komponente berücksichtigt, wo es nützlich erscheint.

Im Konzept selber tauchen für die Umsetzung des Empfehlungssystems wichtige Objekte auf, diese werden als **user** bzw. **item** gekennzeichnet. Zu den Objekten existieren in der Mehrzahl der Fälle notwendige Attribute, welche als *Attribut* dargestellt werden. Optionale Attribute werden als *?Attribut?* gekennzeichnet.
Alle diese "empfehlungswürdigen" Objekte werden in den Abschnitten 4.1, 4.2, 4.3 und 4.4 behandelt und die Empfehlungsgebung wird, soweit nicht schon anderweitig erfolgt, an den entsprechenden Stellen erläutert.

4.1 Einsatzzweck

Dieser Abschnitt dient der Konzeption des Empfehlungssystemes aus Sicht des Einsatzzweckes des Portals *myKOSMOS*. Einmal wird dabei von der Definition des Portals auf seiner Website [kos14] ausgegangen und diese auf Ansatzpunkte untersucht. Eine weitere Sicht beschäftigt sich mit der Konzeption aus Sicht der in Kapitel 1 beschriebenen Kontextorientiertheit oder auch Kontextsensitivität des Portals.

4.1.1 Definition des Portals

Die Definition des Einsatzzwecks dieses Portals ist auf der Website des Projektes *KOSMOS* der Universität Rostock [kos14] zu finden.

Dort ist die Aufgabe des Portals als Allererstes als Unterstützung des sog. *Lebenslangen Lernens* (LLL) definiert. Diese Aufgabe soll durch ein umfangreiches Konzept bewältigt werden, welches unter anderem "[...] Nachfrage, Beratung und Machbarkeit in den Fokus stellt und dabei den Forschungsbezug und die Wissenschaftlichkeit nicht vernachlässigt: nachfrageorientiert durch die Analyse der zielgruppenspezifischen Bedarfe; zukunftsfähig durch ein exzellentes Beratungskonzept, das auch den Menschen einschließt, der abseits des geradlinigen Weges Qualifikationen und Kompetenzen gesammelt hat; [...]" [kos14]. Hier lässt sich bereits der ein oder andere Ansatzpunkt für ein Empfehlungssystem finden. Zum einen findet man den Begriff der Beratung bereits am Anfang dieses Absatzes, dies ist ein mögliches Einsatzgebiet für ein Empfehlungssystem. Des weiteren enthält diese Definition die Aussage der "Nachfrageorientiertheit durch Analyse zielgruppenspezifischer Bedarfe". Durch diese Aussage wird deutlich, dass Bedarfe einer (oder mehrerer) Zielgruppe(-n) erkannt und durch ein geeignetes System entsprechend befriedigt werden sollen. Dies ist ein weiteres Gebiet auf dem ein Empfehlungssystem zielfördernd eingesetzt werden kann. Im letzten Teil dieser Definition wird der Begriff eines "Beratungskonzeptes, welches auch den Menschen einschließen soll, der abseits des gradlinigen Weges Qualifikationen gesammelt hat", erwähnt. Auch hier findet sich ein mögliches Einsatzgebiet für ein Empfehlungssystem, da dadurch Menschen positiv in ihrem Arbeitsfluss und beim Lernen unterstützt werden können.

4.1.2 Kontextsensitivität

Ein weiterer wichtiger Punkt, der im Einsatzzweck des Portals aufgegriffen wird, ist die im Kapitel 1 erwähnte Kontextorientiertheit des Portals. Dies umfasst einerseits die Anpassung von Empfehlungen an den jeweiligen Nutzer, andererseits sollte auch entsprechend die Anpassung des Aussehens, also des Layouts des Portals umfassen. Dieser Punkt wurde bereits in der Arbeit von [Ack14] aufgegriffen.

Dabei ist es einmal denkbar, dem Nutzer gezielt, wie bereits in Abschnitt 2.4 durch das SUPPORT2-PORTLET beschrieben, Portlets für die Individualisierung seiner Benutzeroberfläche (GUI) zu empfehlen. Dies kann mit einem gewissen Aufwand in das Empfehlungssystem des Portals integriert werden. Es ist auch sinnvoll dies zu tun, um etwaige Inkompatibilitäten, sei es aktuell oder durch Aktualisierungen am Portal, dem Portal-Empfehlungssystem oder einer anderen Software, zu verhindern oder abzumindern.
Ebenso ist es sinnvoll, den im Portal arbeitenden Nutzer durch eine Umgestaltung seiner Arbeitsumgebung (im Portal) entsprechend zu unterstützen. Dazu muss lediglich festgestellt werden, in was für einem Kontext sich der Nutzer gerade im Portal bewegt. Dazu sollte eine entsprechende Software mit passender und kompatibler Datenstruktur gefunden werden, die entsprechende Bewegungen und Interaktionen im Portal erfasst, diese Daten speichert ([Ack14] lieferte dazu bereits hervorragende Ansätze) und dann entsprechend diesen das Layout umstellt, sei es durch direkten Zugriff oder Weitermeldung eines Status an eine "Layout-Einheit", die dann entsprechende Umverteilungen vornimmt. Dabei kann grob zwischen drei verschieden Status differenziert werden:

- Status 1: Nutzung des Portals zu privaten Zwecken
- Status 2: Nutzung des Portals zu akademischen Zwecken
- Status 0: Nutzung des Portals zu gemischten Zwecken (sowohl privat als auch akademisch)

Akademische Zwecke umfassen hierbei die Nutzung des Portals z.B. zur Information über Hausaufgaben/Prüfungen, Literaturrecherche, Lesen von universitären Mitteilungen aller Art, etc.. Die privaten Zwecken umfassen z.B. die Nutzung von sozialen Netzwerken, Internetseiten mit persönlich favorisierten Inhalten (Spiele, etc.).

Die Nutzung zu gemischten Zwecken soll in der Aufzählung als “Normalzustand“ bzw. “Startzustand“ verstanden werden, in dem sich die Nutzer zu Anfang, und höchstwahrscheinlich auch im späteren Verlauf der Nutzung befinden werden, weswegen er die Statusnummer 0 erhält (nicht bindend).

Entsprechend dieser ersten Unterteilung sollte jeder Status eine entsprechende Änderung im Aussehen des Portals auslösen.
Dabei sollte im “Normalzustand“ 0 kein spezieller Fokus weder auf private noch auf akademische Funktionen, sprich Portlets, gelegt werden.
Im Status 1 “private Nutzung“ wir der Fokus und das Layout speziell auf die Bedürfnisse für den Zeitvertreib gelegt. Das bedeutet, das Portlets mit akademischem Hintergrund verkleinert oder ganz aus dem Layout entfernt werden, und dafür andere Funktionalitäten, die mehr für das private Arbeiten geschaffen worden sind, in den Vordergrund gerückt werden, sei es durch vergrößerte Darstellung oder dynamisches Hinzufügen zum Layout.
Das Layout für den Status 2 “akademische Nutzung“ wird den Fokus mehr auf das Arbeiten im wissenschaftlichen Hintergrund legen. Dabei werden Portlets mit privaten Funktionalitäten in den Hintergrund gerückt, sprich verkleinert oder entfernt, und Funktionalitäten für akademische Zwecke werden in den Vordergrund geholt.

Ebenso ist es sinnvoll im Layout neben den Funktionalitäten und Portlets auch die Farbgebung entsprechend zu verändern. Dies kann, wie am Beispiel des *Jolla Smartphones* ([jol14a]) durch die sogenannten *The Other Half*s ([jol14b]), im Portal durch die Erkennung des Kontextes des Nutzers geschehen. Sinnvollerweise sollte dabei beim Status 1 “privater Zweck“ ein tendenziell farbenfroheres, d.h. mehrfarbiges, Layout gewählt werden, hingegen ist beim Arbeiten in Status 2 ein eher einfarbiges und dadurch farbneutrales Layout besser, um den Arbeitsfluss nicht zu stören. Im Status 0, dem “Normalzustand“ sollte eine Mischung aus den Farb-Layouts von Status 1 und 2 verwendet werden.
Gleichzeitig ist es sinnvoll, den Nutzer selbst entscheiden zu lassen, welches Farbschema er wählen möchte, wenn sich das Portal um ihn herum verändert, im Sinne der *Jolla Other Half*s.

Ob ein Farbschema komplett für alle 3 (oder mehr) Status verwendet wird oder ob jeder Status sein eigenes Farbschema bekommt, sollte in der Hand des Nutzers selbst liegen, auch wenn dies einen leicht erhöhten Aufwand bei der Verwaltung darstellt. Dies erhöht die Nutzerakzeptanz eher, als wenn ein vorgefertigtes Farbschema vorgegeben wird, da der Nutzer dann für sich eine aktive Mitbestimmung (wenigstens für die Farbe) bei der Personalisierung seiner Benutzeroberfläche feststellen kann.

Ebenso denkbar wäre die Möglichkeit für den Nutzer, statt der beiden oben genannten Verfahren, die automatisch und (fast) ohne Zutun des Nutzers ablaufen, eine manuelle Umstellung des Layouts zu ermöglichen. Diese wäre dann allerdings permanent und würde wiederum Nutzereingabe erforderlich machen, um eine erneute Umstellung auszulösen.

Eine weitere denkbare Variante wäre auch die beiden beschriebenen Verfahren zu kombinieren, in dem man der manuellen Variante bis zu einem gewissen Punkt, in der Beobachtung der Aktivitäten im Portal, den Vorzug gibt und das Layout dann automatisch umstellt. Natürlich muss der Nutzer dann auch selbst bestimmen können, ob er ein Layout dauerhaft haben möchte. Das Empfehlungssystem hingegen sollte, wenn es einen anderen Kontext über eine gewisse Zeitspanne feststellt, der dem aktuelle Eingestellten widerspricht, dem Nutzer eine andere Einstellung vorschlagen. Dies könnte durch einen dezent gesetzten Hinweis geschehen.

4.2 Prozessorientierte Sichtweise

Nachdem im vorangegangenen Abschnitt ein Teil des Konzeptes aus der Einsatzsicht des Portals erarbeitet wurde, soll nun die Konzepterstellung aus Sicht der Prozessorientierung und -modellierung stattfinden.

4.2.1 Modelle

Dazu werden in den folgenden Abbildungen 4.1, 4.2 sowie 4.3 typische Studenten modelliert, die ein oder mehrere Semester durchlaufen, um so Anknüpfungspunkte für ein Empfehlungssystem zu finden. Die Modellierung erfolgt in UML State Chart Diagrammen, dabei wurden auf Beschriftungen von selbsterklärenden Übergängen und auf unnötige Zwischenschritte verzichtet, um die Lesbarkeit zu gewährleisten.

Mit diesen Modellen wurde versucht ein möglichst großes Spektrum an Studenten abzudecken, weswegen diese Modelle als Skizze zu verstehen sind. Punkte im Ablauf der Modelle, an denen eine Empfehlungssystem sinnvoll eingesetzt werden kann, werden mit einem Kreis markiert und nach den Modellen entsprechend erläutert.

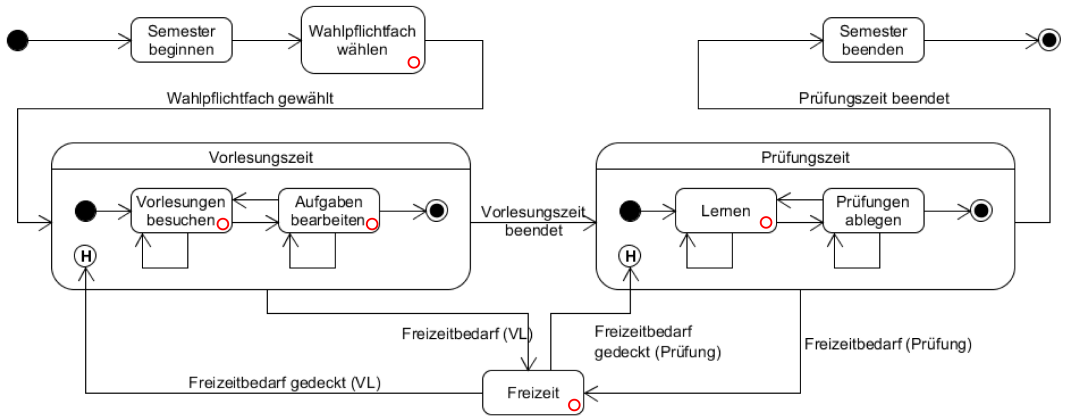

Abbildung 4.1: Student 1

Student 1 bildet einen Studenten in einem Semester ab, in welchem ein Wahlpflichtfach gewählt werden muss. Nachdem dies erfolgt ist, beginnt die “Vorlesungszeit“ in der Vorlesungen besucht und Aufgaben im Semester erledigt werden. Nach dem Ende der “Vorlesungszeit“ beginnt die “Prüfungszeit“, in der für Prüfungen gelernt und diese am entsprechenden Termin abgelegt werden. Aus den komplexen Zuständen ist es jederzeit möglich in den Zustand “Freizeit“ und wieder zurück an die entsprechende Stelle zu wechseln. Nach dem Ende der “Prüfungszeit“ wird das Semester als beendet erklärt.

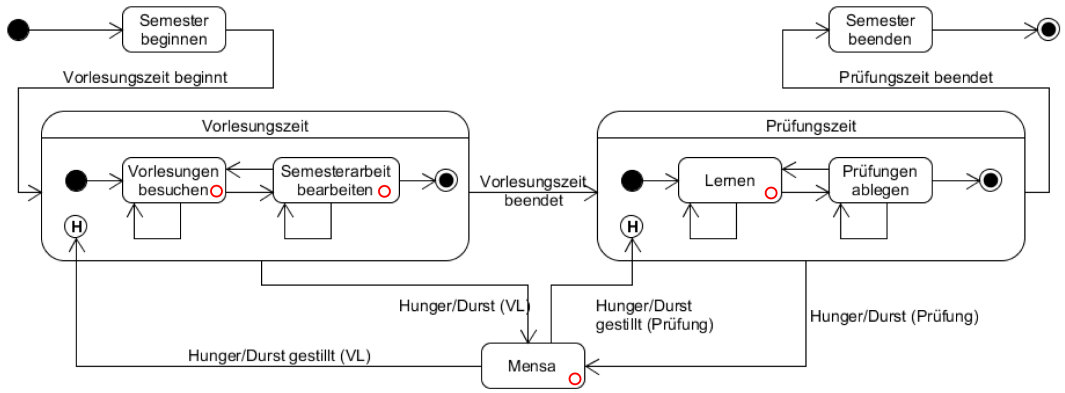

Abbildung 4.2: Student 2

Student 2 hingegen bildet einen Studenten ohne Wahlpflichtfach ab. Der einzige weitere Unterschied zum Modell in Abbildung 4.1 ist, das statt des Zustandes “Freizeit“ ein Zustand namens “Mensa“ eingeführt wurde, welcher stellvertretend für jede Art von gastronomische Einrichtung steht. Die Bezeichnung Mensa war naheliegend, da jedem Studenten der Begriff Mensa vertraut ist und er genau weiß, was damit assoziiert wird.

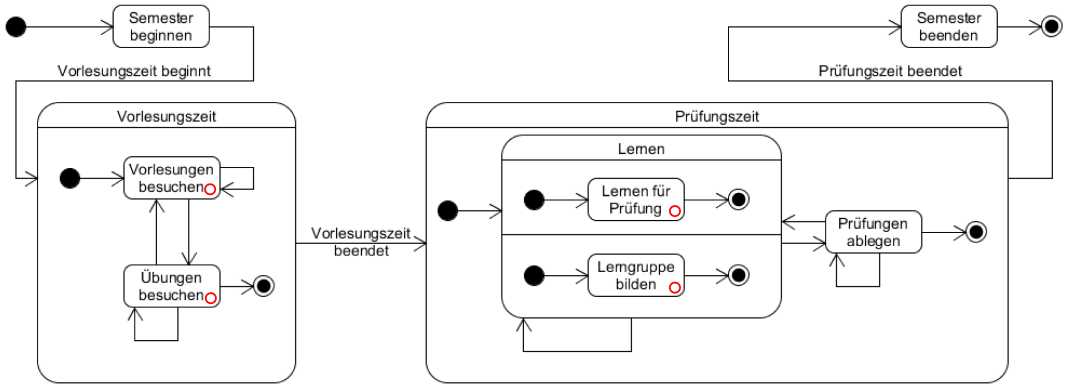

Abbildung 4.3: Student 3

Bei Student 3 wurde der Zustand “Mensa“/“Freizeit“ entfernt, dafür wurde ein weiterer komplexer Zustand in der “Prüfungszeit“ eingeführt, der den Namen “Lernen“ trägt.In diesem Zustand wird parallel zum Lernen für eine Prüfung, eine Lerngruppe aufgebaut.

4.2.2 Identifizierte items

Nachdem die Modelle eingeführt, kurz erläutert und wichtige Punkte im Ablauf eines Semesters in diesen Modellen markiert wurden, sollen im folgenden Text die bei diesen Punkten identifizierten empfehlungswürdigen **items** beschrieben werden. Dabei wird einmal das **item** an sich und die Möglichkeiten der Empfehlungsgebung für dieses **item** beschrieben. Dabei werden aus Gründen der Übersichtlichkeit auch nicht aus den Modellen ersichtliche Zusammenhänge detailliert dargestellt wären.

Prüfungen

Bei den Prüfungen gestaltet sich die Empfehlungsgestaltung relativ einfach, da am Anfang wenig Berechnungsaufwand besteht. Der *Studiengang* des **Nutzers**, inklusive des Prüfungsplanes, ist dem LDAP der Universität Rostock zu entnehmen. Ebenso sind dort bereits angemeldete **Prüfungen** (inklusive des *Prüfer*, der *Uhrzeit* und des *Ortes*) verfügbar, die nur noch von dort extrahiert werden müssen.
Man kann in dieser Sektion durchaus unterscheiden, ob bereits Prüfungen angemeldet wurden oder noch anzumelden sind. Sind Prüfungen noch nicht angemeldet, wäre es sehr sinnvoll den Nutzer gezielt darauf hinzuweisen, das noch **Prüfungsanmeldungen** ausstehen. Natürlich sollte es dem Nutzer stets möglich seine Meldungen abzustellen, sei es dauerhaft oder temporär. Auch ist es möglich, den Nutzer auf bekannte **Prüfungstermine** hinzuweisen. Dies sollte natürlich nicht permanent geschehen, sondern termingerecht zur Prüfung passieren. Zugleich sollte jeder Hinweis zu den Prüfungen so kurz wie möglich, aber so lang wie nötig gehalten werden. Treten mehrere Termine oder Anmeldungsempfehlungen gleichzeitig auf, so sollten diese entsprechend visuell aufbereitet und zusammengefasst werden, um dem Nutzer die Information/en in konzentrierter Form zukommen zu lassen.
An dieser Stelle kann und sollte man die Empfehlungsgebung noch erweitern, in dem zu der Prüfung passende **Literatur**, **Hausaufgaben** und in Zukunft vielleicht **Lerngruppen**, **Dateien** (siehe Abschnitt 4.4) sowie **Websites** empfohlen werden. Die Empfehlung von **Literatur** kann auf Basis der Literaturliste des Dozenten der Vorlesung geschehen, wie auch auf Basis von anderen Nutzern, die zu dieser Vorlesung passende Literatur gesucht und gefunden haben.

Die **Hausaufgaben** müssen für die Empfehlungsberechnung im System sein und die in Abschnitt 4.3 genannten Attribute aufweisen, damit sie korrekt für Empfehlungen verwendet werden können. Für die Empfehlungen von **Lerngruppen** sollten diese ebenfalls Attribute für deren Identifikation aufweisen. Diese sollten die *Mitglieder, Gruppengröße* sowie vielleicht einen *Status*, der angibt, ob die Gruppe geschlossen ist, oder weitere Beitritte möglich sind, enthalten.

Hausaufgaben

Zu den Hausaufgaben lassen sich ebenfalls Empfehlungen berechnen. Hier ist jedoch die Komplexität der Empfehlungsbrechnung vom eigentlichen Zweck der Empfehlung abhängig. Soll die **Hausaufgabe** einfach nur empfohlen werden, weil sie gerade aktuell ist (Ausgabe, Abgabe, Änderungen) oder soll die Hausaufgabe empfohlen werden, weil z.B. für die Prüfungsvorbereitung benötigt wird oder sogar vom Dozenten als *prüfungsrelevant* markiert wurde?
Ist die Empfehlung der Hausaufgabe als einfache Information im Sinne des Bearbeitens der Selbigen gedacht, so ist die Empfehlungsberechnung relativ simpel. Notwendig ist lediglich die Berücksichtigung des Zeitraumes der Aufgabe, d.h. Ausgabe- und Abgabetermin. Die Bearbeitungszeit ergibt sich aus der Differenz der Daten, um so die Hausaufgabe zielgerichtet, das bedeutet termingerecht mit allen notwendigen Informationen, an den Nutzer zu bringen. Die notwendigen Informationen der Empfehlung umfassen für diesen Fall nur die **Hausaufgabe** mit der Aufgabenstellung an sich, den *Bearbeitungszeitraum* und den *Ort* für den Bezug der Aufgabe. Der Ort wäre in diesem Falle eine **Website** (z.B. das ILIAS der Universität Rostock), auf der die Aufgaben gelesen, heruntergeladen und auch gleich gelöst und diese Lösung anschließend auf Korrektheit überprüft werden können. Der Ort kann aber auch beispielsweise zu einem anderen Portal oder Cloud-Dienst führen, wo die **Hausaufgaben** von Dozenten als **Dateien** abgelegt werden können. An jedem Ort aber muss eine Datei für eine gewisse Zeitspanne abgelegt werden können und auch für alle relevanten **user** zugänglich sein.
Im Weiteren wäre es ebenso sinnvoll den Nutzer vor Ende des Bearbeitungszeitraumes auf diesen Umstand hinzuweisen und dadurch aufmerksam zu machen. Ebenso können so gezielt Informationen für die Bearbeitung der Hausaufgaben bereitgestellt werden, sei es eine Änderung an der Aufgabe selber (z.B. Korrektur von Fehlern der Aufgabenstellung) oder für die Bearbeitung nutzbare **Literatur** (kollaborativer Ansatz).

Ist die Empfehlung der **Hausaufgabe** hingegen als Empfehlung im Rahmen der Prüfungsvorbereitung zu verstehen, so verkompliziert sich die Empfehlungsberechnung recht stark, da nun der Fokus eher auf der Interaktion der **Nutzer** mit allen **Hausaufgaben** eines **Prüfungsfaches** liegt und das RCS einen Bezug zwischen den **Nutzern** herstellen muss, um adäquate Empfehlungen geben zu können. Dieser steigenden Komplexität kann aber entgegengewirkt werden, beispielsweise indem der Dozent wichtige **Hausaufgaben** als *prüfungsrelevant* kennzeichnet oder/und eine Übungsserie für die Prüfung erstellt, welche dann mit besonderer *Priorität* empfohlen werden kann. Sollten diese oder andere zusätzlichen Angaben aber nicht vorhanden sein, so muss die Empfehlungsberechnung klassisch geschehen, d.h. ohne zusätzlich Hilfestellungen seitens des Portals und über den kolloborativen Ansatz.
Zudem können wie oben bereits genannt, zu der **Prüfung** oder der **Hausaufgabe** passende **Lerngruppen/-partner** empfohlen werden.

Hausarbeiten/Semesterarbeiten

Mit den **Haus-/Semesterarbeiten** kann genauso verfahren werden, wie mit den **Hausaufgaben**. Dies setzt natürlich eine entsprechende effiziente Datenstruktur voraus. Allerdings sollte beachtet werden, ob diese Arbeiten in bereits festgelegten Gruppen erfolgen, dies muss dann auch entsprechend erfasst werden können. Ebenso könnte es relevant sein, regelmäßig (bei größeren Zeiträumen) über die verbleibende Zeit, die noch zur Verfügung steht, zu informieren.

Übungstermine/-gruppen

Die **Übungstermine/-gruppen** sind ebenfalls eine Gruppe von **items**, welche in einem Empfehlungssystem verarbeitet werden können. Hierbei ist auch wieder der Zweck ausschlaggebend für die Komplexität der Berechnung.
Bei den **Übungsterminen** muss nur der zeitliche Kontext berücksichtigt werden. Weiterhin ist eine Erweiterung der Empfehlungsgebung um die Beachtung des aktuellen Vorlesungsplanes bzw. dessen Struktur sinnvoll, um Terminüberschneidungen zu vermeiden. Die Gruppengröße und die Raumgröße könnten ebenfalls mit in Betracht gezogen werden.

Die Empfehlungen zu den **Übungsgruppen** können wie bei den **Lerngruppen/-partnern** geschehen, weiteres im nächsten Absatz. Ebenso könnte auf Basis von bestehenden Gruppen eine Empfehlung gegeben werden.

Lernpartner/-gruppen

Die bereits erwähnten **Lerngruppen/-partner** stellen ebenfalls **items** da, welche empfehlungswürdig ist. Hierbei ist es weniger ausschlaggebend, zu welchem Zweck z.B. die Gruppe gegründet wurde, sondern eher wie gut die einzelnen Gruppenmitglieder zueinander passen. Da der soziale Faktor eher schlecht maschinell erfassbar ist, muss sich die Empfehlungsberechnung auf die auf die in *Liferay* vorhandene Freundesliste und Nutzerähnlichkeiten bei der Interaktion mit dem Portal stützen, um so zu einem klaren Bild zu gelangen. Dies kann für den Anfang beispielsweise auf Basis des *Studienganges* eines **Nutzers** geschehen. Da dies aber zu einer sehr großen Masse an ähnlichen Nutzern führen würde (gerade bei Studiengängen mit einer hohen Anzahl von Studenten), wäre es sinnvoll, diese Empfehlung später ins jeweilige Semester zu verlegen und erst einmal die Nutzer selber die Einteilung vornehmen lassen, sowohl in der Freundesliste als auch zu Gruppen und zu Partnern. Ebenso kann eine Empfehlung auf Basis von eventuell vergangenen Zusammenarbeiten gegeben werden, diese müssten natürlich erfasst werden. Dazu ist es allerdings notwendig eine Funktion anzubieten, in der Nutzer **Gruppen** und **Lernpartnerschaften** anlegen können.

Extra-Vorlesungen der Dozenten

Zusätzliche Vorlesungen eines Dozenten in einem Semester sind ebenfalls für die Empfehlungsgebung geeignet, da diese meistens wichtige Themen wie z.B. Prüfungsinhalte oder Hilfestellungen für das Verstehen von Zusammenhängen in Hausaufgaben und anderes enthalten. Hierbei ist es wichtig zu beachten, dass diese Extra-Vorlesungen meist einen höheren Status, relativ zu den üblichen Vorlesungen, besitzen, weswegen die Extra-Vorlesungen bei der Präsentation der Empfehlung besonders gekennzeichnet sein sollten. Dies kann beispielsweise durch farbliche Hervorhebung im Vorlesungsplan oder ein zusätzliches Fenster geschehen. Damit die Empfehlungsgebung überhaupt stattfinden kann ist es elementar, dass die Dozenten diese zusätzlichen Vorlesungen auch in eine Datenstruktur eintragen können, die eine solche Kennzeichnung unterstützt.

Die eigentliche Berechnungsempfehlung gestaltet sich in diesem Falle wirklich als sehr einfach, da nur der zeitliche Kontext dieser Extra-Vorlesung beachtet werden muss. Das bedeutet dass die Empfehlung termingerecht gegeben wird und der *Inhalt* sowie die *Zeit* und der *Ort*, und ggf. auch der *Dozent* (sollte eine Abweichung vorliegen), als Informationen angegeben werden.

Wahlpflichtfächer

Wahlpflichtfächer bilden ebenfalls eine Gruppe von **items**, die für eine Empfehlungsgebung geeinigt sind. Bei der Präsentation kann ähnlich verfahren werden, wie im vorherigen Absatz beschrieben wurde. Allerdings ist es in diesem Absatz wichtig, das der Bezug zu den empfehlenswürdigen **items** über ähnliche Nutzer oder Vorlesungen geschieht, was bedeutet, die Empfehlungsgebung ist ein Idealfall für den hybriden Ansatz, da hier sowohl mit Nutzerähnlichkeiten, als auch mit Vorlesungsähnlichkeiten gearbeitet werden kann und muss, was die Wahrscheinlichkeit einer guten Empfehlung signifikant erhöht. Wichtig ist bei dieser Empfehlung ebenfalls der zeitliche Kontext, da es an verschiedenen Fakultäten der Universität Rostock unterschiedliche Zeitpunkte (sprich Semester) für die Wahl von Wahlpflichtfächern gibt. Die erhöht natürlich die Schwierigkeit der Empfehlungsgebung, da z.B. im Studiengang Physik bereits im ersten Semester Wahlpflichtfächer gewählt werden müssen, wohingegen es im Studiengang Wirtschaftsinformatik erst im dritten Semester geschieht. Dies verkompliziert die Empfehlungsberechnung für das erste Semester massiv, da zu diesem Zeitpunkt kaum Daten über die Studiengangsteilnehmer vorliegen. Zudem dünnt sich der Anteil der weiter Studierenden erfahrungsgemäß im Laufe des ersten Semester (und in späteren) aus, was zu Problemen bei der Berechnung von Nutzerähnlichkeiten führen kann, wenn beispielsweise 4 von 5 aggregierten Studenten ausscheiden.

Vorlesungen anderer Dozenten außerhalb des Studienplanes

Mit den Vorlesungen, die standardmäßig nicht zum Studienplan eines Faches gehören, kann ähnlich verfahren werden, wie im Absatz der Extra-Vorlesungen beschrieben. Diese nicht im Studienplan stehenden Vorlesungen bieten einen hervorragenden Ansatz für ein hybrides Empfehlungssystem. Für diese Vorlesungen muss eine Datenstruktur geschaffen werden, die es erlaubt, zwischen “normalen“ Vorlesungen, Extra-Vorlesungen, Wahlpflichtfächern und zusätzlichen Vorlesungen/Veranstaltungen zu unterschieden.

Sonstige Veranstaltungen im Semester

Ebenso bilden **sonstige Veranstaltungen** eine Gruppe von empfehlungswürdigen **items**. Hierbei sollte unterschieden werden, ob diese sonstigen Veranstaltungen einen Bezug zum Studiengang des jeweiligen Nutzers haben oder nicht. Um einen Bezug herzustellen, kann bereits der Ersteller dieser Veranstaltung dies vorgeben, indem die Priorität bestimmter Studiengänge höher gesetzt wird oder aber eine bestimmte Fakultät bzw. Studiengang oder mehrere als Bezugsgruppe ausgewählt werden, da andere Fakultäten bzw. Studiengänge nicht mit einbezogen werden sollen. Dies ist bei fakultäts-/studiengangsinternen Abläufen relevant. Eine andere Möglichkeit bzw. eine zusätzliche Option um einen Bezug zu bilden, wäre der Einsatz eines inhaltsbasierten Empfehlungssystems, welches die Beschreibung der Veranstaltung durchsucht und daraus Ähnlichkeiten berechnet. Ob der Einsatz einer kollaborativen Komponente, also die Erweiterung auf ein hybrides Empfehlungssystem sinnvoll ist, lässt sich an dieser Stelle nur schwer abschätzen. Auf der einen Seite könnte es sinnvoll sein, da einige Studenten aktiver nach solchen Veranstaltungen suchen als andere. Auf der anderen Seite wäre es möglich, dass sich trotz dieser Studenten zu wenige brauchbare Daten ergeben. Dies sollte an geeigneter Stelle evaluiert werden. Natürlich muss für diese Zwecke eine Datenstruktur existieren, welche die genannten Möglichkeiten abdeckt.
Sollte ein solcher Bezug vorliegen, so sollte die Empfehlung entsprechend gegeben und präsentiert werden. Ohne einen konkreten Bezug, beispielsweise bei einer universitätsweiten Veranstaltung (z.B. HIT), wäre es sinnvoll, dies entsprechend bei der Empfehlungsgebung kenntlich zu machen.

Suche

Bei der **Suche** allgemein kann ein Empfehlungssystem durch Empfehlungen unterstützend wirken. Auf das Portal bezogen wären demzufolge spezielle Empfehlungen für das akademische Arbeiten mehr als sinnvoll. Wichtig dabei ist, dass ausgewertet werden kann, in welchem Kontext sich der Nutzer gerade durch das Portal bewegt, sei es nun privater oder akademischer Natur. Dabei kann man zwischen drei Möglichkeiten der Empfehlung grob unterscheiden, auf der einen Seite kann man dem Nutzer nur eine **Website** oder einen *Ablageort* empfehlen, das bedeutet man gibt dem Nutzer nur eine grobe Richtung vor.

Diese Variante wäre mehr für den Beginn der Laufzeit des Portals geeignet, da man hierbei z.B. auf eine festgelegt Liste von **Websites** zugreifen kann, welche ausschließlich akademische Seiten enthält, wie z.B. die Springer-Verlags-Seite, Google Scholar oder ai-sel.org, wenn die Nutzerbeobachtung noch keine ausreichenden Ergebnisse für eine gute Empfehlungsgebung liefert. Für eine größer Menge an Nutzerdaten, die sich zwangsläufig mit der Alterung des Protals ergibt, wäre es denkbar, Nutzern auf Basis der hybriden Methode gleich einen konkreten Teil einer **Website** oder eine konkrete **Datei** zu empfehlen.
Eine zweite Möglichkeit ist, dass das Empfehlungssystem den Nutzer bei der Eingabe von Suchbegriffen unterstützt, im Sinne der Auto-Vervollständigung von Suchmaschinen, hier sei als ein populäres Beispiel die Methode von Google in Abbildung 4.4 gezeigt. Im Portal sollte die Präsentation ähnlich erfolgen, nur ist es für *myKOSMOS* nicht notwendig, wie in den verpixelten Bereichen in Abbildung 4.4 Werbung zu platzieren.

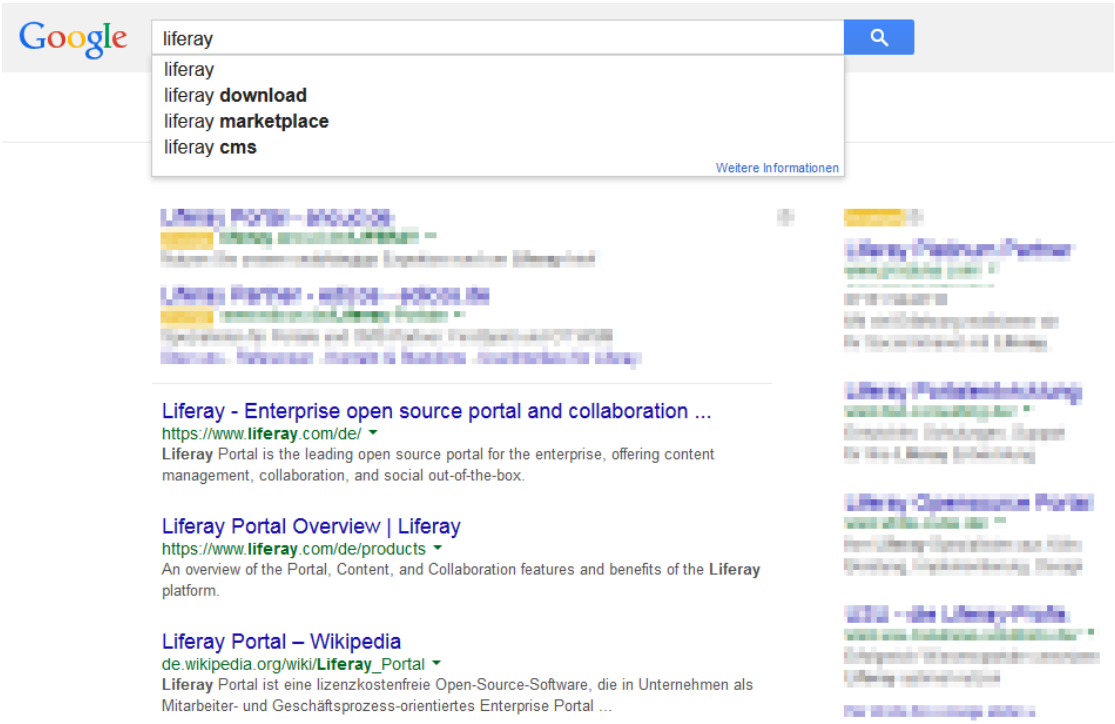

Abbildung 4.4: Google Auto-Vervollständigung

Auch hierbei kann ein Empfehlungssystem hilfreich sein, da sich hier ein hervorragender Ansatz für die hybride Methode finden lässt. Auf der einen Seite können über den kollaborativen Ansatz andere *Suchbegriffe* von ähnlichen Nutzern übernommen werden, dies setzt natürlich voraus, dass Suchen von anderen Nutzern effizient abgespeichert und am besten sogar aggregiert werden, um schnell Empfehlungen generieren zu können. Dabei könnte man auch den Weg von quasi-statisch Empfehlungen gehen, das bedeutet, man berechnet Empfehlungen auf Basis eines bestimmten Zeitpunktes und speichert diese dann bis zur nächsten Berechnung zwischen. Dies würde die Last auf das Empfehlungssystem minimieren, da nun nicht in Echtzeit, sondern sprichwörtlich Empfehlungen "auf Vorrat" generiert werden können, die nur für eine bestimmte Zeitspanne gültig sind.
Auf der anderen Seite können durch die inhaltsbasierte Variante der Empfehlungsberechnung vom Suchbegriff aus ähnliche Begriffe errechnet werden. Auch dies kann wahlweise entweder in Echtzeit oder "auf Vorrat" geschehen. Auch hier muss eine effiziente Datenstruktur zur Abspeicherung entworfen werden, damit es zu keinen Verzögerungen durch Abfragen der Daten kommt.
In einer dritter Möglichkeit wäre es denkbar, die Suchergebnisse selbst auf Basis der Popularität nach der hybriden Methode zu verändern, sprich der Ähnlichkeit von Ergebnissen und deren Popularität bei anderen Nutzern und deren Ähnlichkeit mit dem jeweiligen Nutzer der gerade sucht. Die "Wegtam-Suche" bietet dazu bereits Möglichkeiten.

Räume/Raumpläne

Die Empfehlung von **Räumen** unter Berücksichtigung von **Raumplänen** stellt ebenfalls ein weiteres Anwendungsgebiet für ein Empfehlungssystem da.
Raumempfehlungen könnten beispielsweise gegeben werden, wenn ein Nutzer in einem Gebäude der Universität Rostock mit dem Portal arbeitet und das Portal nun eine Nutzung zur Prüfungsvorbereitung feststellt. Dies sollte ebenfalls durch eine manuelle Suche nach verfügbaren Räumen geschehen. Dann könnte das Portal eine Empfehlung ausgeben, in welcher der Nutzer auf einen freien Raum in seiner Nähe aufmerksam gemacht wird.
Dafür ist es notwendig, das alle öffentlich zugänglichen Räume erfasst werden, inklusiver der Besetzung dieses Raumes während des Semesters (**Raumplan**), eventueller zusätzlicher Ausstattungsmerkmale sowie eventueller Zugangsbeschränkungen (ITMZ-Karte).

Dies wäre wichtig, um nicht einen Raum zu empfehlen, der eventuell nur 2 Stunden am Tag verfügbar ist und bereits nach 20 Minuten durch eine Vorlesung oder Übung, die in diesem Raum stattfindet, wieder belegt wird. Diese Pläne sollten in der Verwaltung der Universität Rostock vorhanden sein und müssen dann nur in eine geeignete Datenstruktur übernommen bzw. überführt werden. Dabei ist es wichtig, dass auf eventuell besetzte Räume entsprechend reagiert werden kann. Bei festen Terminen stellt dies kein Problem dar, anders hingegen gestaltet es sich, wenn der Raum außerhalb des Planes, beispielsweise im Rahmen einer Lerngruppe besetzt wird. Hier muss eine Möglichkeit geschaffen werden, auf eventuelle außerplanmäßige Nutzer zu reagieren, in dem sie überhaupt wahrgenommen werden können. Dies könnte beispielsweise durch die Beobachtung von WLAN- oder Stromaktivitäten (Licht, Computer, Steckdosen) eines Raumes geschehen. Ebenso wäre es wichtig den Nutzer bei der Empfehlungsgebung vor einer Neubesetzung des Raumes entsprechend zu warnen, sei es zeitnah oder gleich bei der Empfehlung.
Ebenso könnte diese Empfehlungskomponente generell bei der Raumplanung in den entsprechenden Stellen eingesetzt werden, um so den reibungslosen Ablauf zu gewährleisten.

Mensa, generell Gastronomie

Auch bei den gastronomischen Einrichtungen der Universität Rostock, der Mensa, bzw. Mensen des Studentenwerkes kann man von empfehlungswürdigen **items** sprechen.
Speisepläne sind online verfügbar, ebenso existieren bereits einige Apps für Smartphones wie z.B. die “CampusApp“ oder “Mensaplan“, die den Speiseplan der ortsansässigen Mensen entweder als Teilfunktion (Campus-App) oder als alleinige Funktion (Mensaplan) anbieten. Die große Schwierigkeit bei der Empfehlungsgebung ist in diesem Zusammenhang natürlich der persönliche Geschmack eines jeden Menschen. Deswegen ist ein erster Vorschlag, nicht direkt Speisen zu empfehlen, sondern die Nutzer eher darauf aufmerksam zu machen, dass die Mensa gerade geöffnet hat (, in Kürze öffnen/schließen wird oder dauerhaft geschlossen ist (z.B. vorlesungsfreie Zeit)), wo sich die nächstgelegene Mensa befindet (OpenStreetMaps), welche Preise für welches Gericht anfallen, etc.. Hierbei ist es für die Empfehlungsgebung wichtig, den zeitlichen Kontext und vor allem den Nutzerkontext zu erkennen. Dies kann durch Beobachtung des Verhaltens geschehen, wenn das System feststellt, das der Nutzer sich über die aktuellen Gerichte der Mensa informiert.

Dies könnte beispielsweise durch ein eigenes Portlet geschehen, welches die am Tag jeweils aktuellen Gerichte darstellt. Dies würde die Erstellung eines Profiles begünstigen und vereinfachen, da der Nutzer nun eine einfache Alternative zur Suche im Internet oder per App hat, die auch vergleichsweise einfach überwacht (z.B. durch Klick-zählung) werden kann.
Um die Empfehlung nun um konkrete **items**, also **Gerichte** zu erweitern, muss ein Nutzerprofil vorliegen. Dieses lässt sich am einfachsten durch Bewertungen von Speisen erstellen, da man dort mittels des hybriden Ansatzes über Ähnlichkeiten der **Gerichte** zu schnellen Ergebnissen kommen sollte. Ebenso wäre es denkbar, den Nutzer durch das in Abschnitt 4.7 zum Schluss näher beschriebene Verfahren des sogenannten "Canvas Fingerprinting", kurz durch Anlegen eines Bewegungsgraphen auf einer Seite, zu überwachen, um so ebenfalls schneller zu einem Profil zu gelangen.
Generell ist es einfach möglich, die Empfehlungsgebung um zusätzliche gastronomischen Einrichtungen zu erweitern, wie etwa Restaurants, Kneipen/Bars oder auch Lieferdienste. Dazu müssen diese lediglich in ein und dieselbe Datenstruktur eingepflegt werden.

4.2.3 Zusammenfassung

An dieser Stelle werden nun alle aufgetretenen empfehlungswürdigen **items** zusammengetragen und noch einmal betrachtet.

Prüfung

Prüfungen gehören mit zu den wichtigsten empfehlenswerten **items**, da sie einen wichtigen Punkt im Semester und generell im Studium darstellen. Hier kann der Student durch ein Empfehlungssystem optimal unterstützt werden, sei es bei der Anmeldung oder beim Lernen. Für den Einsatz zur Unterstützung des Lernprozesses wäre der hybride Ansatz geeignet.
Die notwendigen Daten für Prüfungen sollten aus dem LDAP der Universität Rostock beschaffbar sein.

Literatur

Unter Literatur wird in dieser Arbeit jedes physikalische Druckerzeugnis aufgefasst, das umfasst Bücher, Magazine, Zeitschriften, etc.; elektronische digitalisierte Varianten davon werden den Dateien bzw. Websites zugeordnet. Auch bei der Literaturempfehlung wird der hybride Ansatz als der Sinnvollste angesehen.
Die Daten hierfür sollten in der Verwaltung der Universitäts-Bibliothek zu finden sein.

Hausaufgaben

Auch Hausaufgabe markieren wichtige Punkte im Semester eines Studenten, seien es wöchentliche Hausaufgaben oder Semesterarbeiten. Auch hier dürfte der hybriden Ansatz erfolgsversprechend sein. Ebenso müssen Hausaufgabe natürlich in einer Datenstruktur erfassbar sein.
Die Daten für die Hausaufgabe müssen von den jeweiligen Dozenten erstellt und dabei effizient eingepflegt werden.

Lerngruppen

Auch Lerngruppen gehören zu dem empfehlungswürdigen **items** in einem Lehr-Lern-Portal. Für diese Art der Empfehlungen muss auch hier eine effiziente Datenstruktur geschaffen werden. Empfehlungen sollten mit dem kollaborativen Ansatz errechnet werden. Die ersten Einteilungen sollten die Studenten selbst vornehmen.
Mit den **Lernpartnern** kann ähnlich verfahren werden.

Vorlesungen

Vorlesungen gehören wie die Prüfungen zu wichtigsten empfehlungswürdigen **items**, da man hier dne Nutzer beispielsweise bei der Wahl von Wahlpflichtfächern durch Empfehlungen positiv unterstützen kann, ebenso bei Extra-Vorlesungen im Semester selbst. Hierfür ist der hybriden Ansatz am besten geeignet.
Die Vorlesungsdaten sind im LSF der Universität Rostock vorhanden und müssen von dort extrahiert werden.

Übungstermine/-gruppen

Übungstermine spielen in Verbindung mit Vorlesungen im Ablauf des Semesters, besonders für die Prüfungsvorbereitung eine große Rolle. Auch hier kann ein Empfehlungssystem den Studenten optimal unterstützen. Hier sollte eine erweiterte Datenstruktur geschaffen werden, um zusätzliche Funktionen zur Übungsplanung beispielsweise aus dem StudIP übernehmen zu können.
Die notwendigen Daten sind sowohl im LSF als auch im StudIP vorhanden.

Dateien

Dateien stellen ebenfalls eine weitere Gruppe von empfehlungswürdigen **items** dar, welche durch Empfehlungssysteme einem Studenten nutzbringend empfohlen werden können. Dies kann zu unterschiedlichen Anlässen geschehen und sollte nach der hybriden Methode geschehen.

Eine Datenstruktur für Dateien ist durch das "My Google Drive Quick Links"-Portlet rudimentär vorhanden und sollte entsprechend der späteren Vorschläge in den Abschnitten 4.3 und 4.4 erweitert bzw. neu aufgebaut werden.

Websites

Auch Websites stellen **items** dar, welche empfohlen werden können. Dies sollte wie beschrieben, durch die Beobachtung des Browsers im Portal geschehen. Die Empfehlung sollte am besten über den kollaborativen Ansatz erfolgen. Ebenso muss eine effiziente Datenstruktur für die Verwaltung geschaffen werden.

Suchen

Auch Suchen können durch ein Empfehlungssystem unterstützt werden, indem entweder passende Suchbegriffe empfohlen werden oder das Suchergebnis umsortiert bzw. gefiltert wird. Hierzu sollten die Möglichkeiten der "Wegtam-Suche" genutzt werden. Empfehlungen können für Suchbegriffe durch den inhaltsbasierten Ansatz auf Vorrat berechnet werden; der kollaborative Ansatz kann für das Filtern der Suchergebnisse genutzt werden.

Raum

Räume stellen ebenfalls eine Gruppe von **items** dar, welche durch ein Empfehlungssystem entsprechend bearbeitet werden können. Hierbei geht es vorrangig darum, einem potentiellen Nutzer der Universität Rostock die Möglichkeit zu geben, nach aktuell unbesetzten Räumen zu suchen bzw. ihm solche zu empfehlen, damit er effektiver arbeiten kann. In Verbindung mit den Raumplänen können so differenzierte Empfehlungen gegeben werden.

Raumlisten sollten in den Verwaltungen zu erhalten sein.

Gastronomische Einrichtungen

Ebenso stellen gastronomische Einrichtungen eine Menge von empfehlungswürdigen **items** da. Diese könne nach dem hybriden Ansatz empfohlen werden, wozu allerdings eine effiziente Datenstruktur erstellt werden muss.

4.3 Technische Ebene

Auf der technischen Ebene des Konzeptes wird zuerst ein Blick auf die bereits in *Liferay* nach einer Neuinstallation verfügbaren Portlets und Datenstrukturen geworfen. In diesem Kontext werden daran anschließend die bisher entwickelten Portlets und Datenstrukturen für "myKOSMOS" genauer untersucht. Im vorletzten Teil dieses Abschnittes wird an die für das Portal entwickelten Portlets anknüpfend die Thematik der "LifeRay-Hooks" erläutert, diese sind als Beobachtungskomponente für das spätere Empfehlungssystem wichtig. Der letzte Teil dieses Abschnittes wird sich mit den im vorangegangenen Abschnitt 4.2 identifizierten **items** beschäftigen, dabei werden Datenstrukturen entworfen, die ein effizientes Arbeiten des zukünftigen portalweiten Empfehlungssystem ermöglichen. Abschließend für diesen Abschnitt wird am Ende eine Zusammenfassung gegeben.

4.3.1 Liferay Grund-Installation

Der technische Aspekt des Konzeptes ist recht umfangreich, weswegen die Beschreibung der *LifeRay*-Neuinstallation nur mit den wichtigsten Portlets und Datenstrukturen geschehen wird, da mit der ausführlichen Beschreibung aller Komponenten ein eigenes Buch gefüllt werden kann. Zudem wäre der zeitliche Aufwand für Einarbeitung und Erläuterung im Rahmen dieser Arbeit nicht mehr vertretbar.

Datenbank

Als Erstes werden nun die Datenstrukturen untersucht, welche nach einer Neuinstallation, sprich als Grundinstallation, vorliegen. Tabellengruppen (≥ 2 Tabellen) werden wie folgt dargestellt: `Tabelle*`. In diesem Teil des Abschnittes wird das Augenmerk mehr auf für ein Empfehlungssystem nutzbare Tabellen gelegt.

Die Tabellen `announcements*` enthalten Daten über Ankündigungen, deren Art und Priorität und dem Weg, wie diese Ankündigungen verbreitet werden oder wurden. Dies könnte bereits von einem Empfehlungssystem zum Empfehlen und auch Darstellen von beispielsweise Veranstaltungsempfehlungen genutzt werden.

Die Tabellen `blogs*` enthalten selbsterklärend Daten über Blogs, deren Inhalt und Bewertung, auch dies könnte von einem Empfehlungssystem genutzt werden, am sinnvollsten über den hybriden Ansatz.
Ebenso ist "Liferay" als Portal in der Lage, Lesezeichen in den Tabellen `bookmarks*` abzuspeichern; dies könnte für die Empfehlung von konkreten Seiten genutzt werden.
Die Tabellen `calendar*` mit `calevent` könnten für die (zusätzliche) Darstellung von Empfehlungen für z.B. Veranstaltungen direkt im Kalender genutzt werden.
Für **Dateien** scheint es ebenfalls eine Struktur zu geben, diese ist in den Tabellen `dlfile*` und Nebenstehenden verankert.
Nutzer haben die Möglichkeit Artikel zu schreiben, diese werden in den Tabellen `journal*` abgespeichert. Auch dies könnte potentiell vom Empfehlungssystem genutzt werden, in dem z.B. über den inhaltsbasierten Ansatz Ähnlichkeiten zu anderen gelesenen Texten berechnet werden.
Für die Zusammenarbeit bietet '"Liferay" die Erweiterung "Kaleo" an, diese wird in den Tabellen `kaleo*` abgebildet. "Kaleo" ist von "Liferay" dazu konzipiert worden, dass die Zusammenarbeit, genauer den Arbeitsfluss, von Nutzern effektiv unterstützt werden kann, in dem Arbeitsgruppen gebildet werden können. [lif15f]
Die Tabellen `mb*` sind notwendige Datenstrukturen für das "Message Board Portlet", das Portalforum in "Liferay" [lif15d]. Dieses Forum könnte ebenfalls für das Empfehlungssystem nutzbar sein, da beispielsweise Lösungen für Aufgaben diskutiert werden könnten, was zu einer Empfehlung für nicht-teilnehmende Nutzer führen könnte. Dies müsste durch einen Hook beobachtet werden.
Die `opensocial*`-Tabellen könnten ebenfalls von einem Empfehlungssystem genutzt werden, um zum Beispiel über kleine sogenannte Gadgets (Portlets in kleinerer Form) Empfehlungen zu präsentieren (näheres in Abschnitt 4.6).
Auch verfügt "Liferay" über die Möglichkeit Umfragen zu gestalten, diese werden in den `polls*`-Tabellen abgespeichert. Dies könnte genutzt werden um beispielsweise einer Empfehlung eine Bewertung zu geben (näheres in Abschnitt 4.7).
Die Tabellenstruktur `quartz_*` dient der Ausführung von chronologischen Hintergrundprozessen in '"Liferay"[lif15g]. Dies könnte beispielsweise für das regelmäßige Berechnen von Empfehlungen zu bestimmten Zeiten dienen.
Die Tabellen `ratings*` sind für alle Bewertungen im Portal zuständig, wobei `ratingsstats` eine aggregierte Übersicht über Bewertungen liefert.

Diese Struktur kann hervorragend vom kollaborativen Ansatz des Empfehlungssystem genutzt werden.
Die `usergroup*`-Tabellen sind ebenfalls hervorragend für das Empfehlungssystem nutzbar, da hier bereits Nutzer zu Gruppen aggregiert worden sind, was die Berechnung zumindest zum Teil vereinfacht.
Die Tabellen `wiki*` bilden nun den Abschluss, in ihnen wird das Wiki des Portals gespeichert, welches bereits in der Arbeit von [Ihl14] für ein Empfehlungssystme genutzt wurde.

Portlets

Nachdem die Tabellen der Grundinstallation abgehandelt wurden, soll der Fokus nun auf die Portlets der Grundinstallation gerichtet werden. Dabei spielen vorrangig die für ein Empfehlungssystem nutzbaren Portlets eine Rolle. Eine Liste aller Portlets ist unter [lif15e] zu finden, diese wird für diesen Teil des Abschnittes chronologisch abgearbeitet.

Unter der Rubrik "Web Publishing" lassen sich bereits erste, für ein Empfehlungssystem nutzbare, Portlets finden.
Zu diesen gehören "Web Content" und "Asset Publisher", in diesen Portlets könnten Empfehlungen dargestellt werden. Ebenso wäre es möglich, aus diesen Portlets durch den hybriden Ansatz (setzt Bewertungsmöglichkeiten voraus) Daten für Empfehlungen zu generieren. Durch die "Tag Cloud", "Tags Navigation" sowie "Categories Navigation" wäre es möglich, dem Nutzer entweder spezifische Schlagwörter oder Kategorien zu empfehlen.
Unter "Document and image management" ist die "Document Library" für ein Empfehlungssystem nutzbar, die Darstellung der Empfehlung könnte als spezifischer Pfad in dem zugehörigen Portlet "Document Library Display" stattfinden. Die "Image Gallery" könnte über zu einem Bild gehörige "Tags" Empfehlungen berechenbar machen.
Die "Web Site Tools" umfasst einige für eine Empfehlungssystem interessante Portlets. "Nested Portlet" würde eine weitere Form der Präsentation von Empfehlungen gestatten, direkt im Portlet als "Unter-Portlet". "Page Ratings" bietet die Option, Seiten im Portal zu bewerten, hier wäre der kollaborative Ansatz für die Berechnung von Empfehlungen geeignet. Hinter "Directory" verbirgt sich die Möglichkeit der Suche nach Nutzern, Nutzergruppen und Organisationen. Auch hier könnte ein Empfehlungssystem ansetzen, um

die Ergebnisse der Suche vom Suchenden unbemerkt zu verändern.

Eine weitere große Rubrik umfasst die “Collaboration Tools“ von “Liferay“.
“Blogs“ wurden bereits im vorangegangenen Teil dieses Abschnittes identifiziert, hier könnten spezifische Einträge über den hybriden Ansatz empfohlen werden. Der “Calendar“ wurde ebenfalls bereits identifiziert, hier könnten Empfehlungen für Veranstaltungen bzw. Vorlesungen erfolgen. Auch die “Message Boards“ als das Portalforum, wurde als nutzbare Datenstruktur identifiziert, hier könnten konkrete Threads oder Kategorien von Beiträgen empfohlen werden. Die Umfragen “Polls“ wurden ebenfalls als für ein Empfehlungssystem nutzbare Struktur identifiziert. Das “Wiki“ wurde bereits in der Arbeit von [Ihl14] als eine Sammlung von empfehlbaren **items** identifiziert. Auch die “Annoucements“ wurde als für ein Empfehlungssystem nutzbare Datenstruktur identifiziert, hier könnten Empfehlungen dargestellt werden. Interessant ist ebenfalls das “Invitation Portlet“, hiermit können Nutzer zu einer Seite im Portal eingeladen werden, dies könnte durch Empfehlungen von “passenden“ Nutzern unterstützt werden.

Die nächste Rubrik trägt den Titel “Social Networking“. In diese Rubrik lassen sich Portlets finden, die Ähnlichkeiten mit den Funktionen bekannter “sozialer Netzwerke“, wie beispielsweise Facebook, haben.
Das “Wall“-Portlet emuliert die Funktionen der Pinnwand von Facebook, hier könnten einerseits Empfehlungen dargestellt werden, anderseits könnten hier ebenso Daten von einer Beobachtungskomponente gesammelt werden, welche dann für die Berechnung von Empfehlungen, beispielsweise von Posts genutzt werden können. Ebenso könnte das “Activities Portlet“ von einem Empfehlungssystem genutzt werden, in dem dort Empfehlungen dargestellt werden können. Das “Friends Portlet“, ebenso wie das ’"Friends Activities Portlet“, können für die Berechnung und Präsentation von Empfehlungen genutzt werden. Das “Friends Portlet“ vereinfacht die Aggregation zu einer Nutzergruppe erheblich, da hier bereits eine Einteilung vorgenommen wurde. Das ’"Members Portlet“ bzw. “Members Activites Portlet“ können ähnlich verwandt werden, diese werden aber auf Organisationsebene und nicht privat angewandt.

Die Rubrik des “e-Commerce“ ist sehr klein, sie wäre dennoch über das “Shopping Portlet“, so denn ein Shop/Store integriert werden würde, für ein Empfehlungssystem nutzbar,

da hier Empfehlungen für **Einkaufsgüter** gegeben werden könnten (siehe Abschnitt 4.4).

Aus den weiteren Rubriken erscheinen lediglich die Portlets “IP Geocoder“ und “Weather“ aus der Rubrik “Other“ nutzbar.
Das “Weather“ Portlet könnte einerseits nur für sich genutzt werden, anderseits könnte es in Verbindung mit dem “IP Geocoder“ (und damit auch mit dem “Maps“ Portlet) für Empfehlungen von "**Räumen** oder **gastronomischen Einrichtungen** genutzt werden.

4.3.2 Liferay *myKOSMOS*

Die für das Portal entwickelten und sich zum Zeitpunkt dieser Arbeit in der Entwicklung befindlichen Portlets enthalten stellenweise nutzbare Datenstrukturen. Inwiefern sich diese für die Arbeit eines Empfehlungssystems eignen, soll in diesem Abschnitt genauer untersucht werden.

Das “Request’ oder “Wegtam“-Portlet enthält leider keine eigenen brauchbaren Datenstrukturen für ein Empfehlungssystem, allerdings können durch Hooks Eingabedaten und Suchergebnisse überwacht werden. Diese Daten können dann von einer Beobachtungskomponente erhoben und in eine geeignete Datenstruktur überführt werden, wo sie anschließend von einem Empfehlungssystem für die Optimierung der Suche für einzelnen Nutzer und Nutzergruppen genutzt werden können. Hierzu sollte, wie bereits erwähnt, die hybride Methode genutzt werden.

Die “StudIP-Anbindung“ verhält sich ähnlich dem “Request“-Portlet. Das “StudIP“-Portlet kann nach dem Stand dieser Arbeit dazu genutzt werden, Ankündigungen über beispielsweise Kolloquien oder sonstige Veranstaltungen priorisiert durch ein Empfehlungssystem an den Nutzer weiterzuleiten. Eine zukünftige Beobachtungskomponente muss in der Lage sein, diese Ankündigungen effizient zu erfassen. Ein Empfehlungssystem kann dann die Auswertung (Empfehlung) dieser Ankündigungen übernehmen. Eigene Datenstrukturen sind zum Zeitpunkt dieser Arbeit nicht bekannt.

Etwas anders verhält es sich mit dem “My Google Drive Quick Links“-Portlet. Dieses Portlet bringt zwar eine einfache Datenstruktur mit sich, diese ist aber durchaus für Empfehlungssysteme nutzbar.

Diese Datenstruktur enthält die Verknüpfungen von Nutzern mit ihren “Google Drive“-Konten, genauer werden dort Nutzern mit Dokumenten (*ID, Name*) und ihrem Ablageort (*URL*) verknüpft. Dies könnte von einer Beobachtungskomponente dazu genutzt werden, Aktivitäten und Interaktionen mit diesen Dateien zu überwachen. Ein Empfehlungssystem könnte aus diesen Daten dann konkrete Dateien oder Ablageorte über den kollaborativen Ansatz empfehlen, ebenso wie über den Inhaltsbasierten.

Das “Skype Chat & IM“ verfügte bereits in seiner Grundform über Datenstrukturen um Skype-Gruppen abzuspeichern, in der erweiterten Form von [DVWG14] wurden andere Strukturen für die Erfassung hinzugefügt. So verfügt dieses Portlet nun über die Möglichkeit, Nutzergruppen zu erstellen und abzuspeichern. Diese Daten können bereits von einem Empfehlungssystem genutzt werden, um Empfehlungen für Nutzergruppen zu bilden. Ebenso ist ein Hook vorhanden, der Aktivitäten in den Gruppen wahrnehmen kann und diese abspeichert. Diese Daten können dann in einem weiteren Portlet dargestellt werden. *Liferay* bietet mit seinem “Activities Portlet“ bereits die Möglichkeit an, portalweit Nutzeraktivitäten zusammenzufassen und darzustellen. Nun stellt sich die Frage, inwieweit die Entwicklung eines zusätzlichen Statusportlets von [DVWG14] notwendig ist und ob man diese Entwicklung nicht in das “Activities Portlet“ integrieren kann, um so nicht unnötig viele Portlets mit ähnlichen oder gar gleichen Funktionen zu haben, die den Nutzer unnötig mit zusätzlichen Informationen überlasten.

Abschließend soll das “Support2“-Portlet untersucht werden. Dieses Portlet beinhaltet neben einer Empfehlungskomponente auf Basis von Regeln (siehe dazu [Ack14]) noch eine Beobachtungskomponente, die über eine eigene Datenstruktur verfügt. Diese Datenstruktur (siehe Abbildung 4.5) wird mit Nutzeraktivitäten des Portals gefüllt, dies geschieht über die bereits erwähnten Hooks, dazu mehr im nächsten Abschnitt. In dieser Datenstruktur können bereits viele wichtige Daten erfasst werden, welche von einem Empfehlungssystem später für die Berechnung genutzt werden können. Dazu gehören u.a. Applikationen (Portlets), Gruppen, Nutzer, Prüfungen. Auf Basis dieser Datenstruktur sollte die weitere Entwicklung einer Empfehlungs- und auch einer Beobachtungskomponente für das Portal geschehen, da hier bereits hervorragende Grundlagen dafür gelegt wurden.

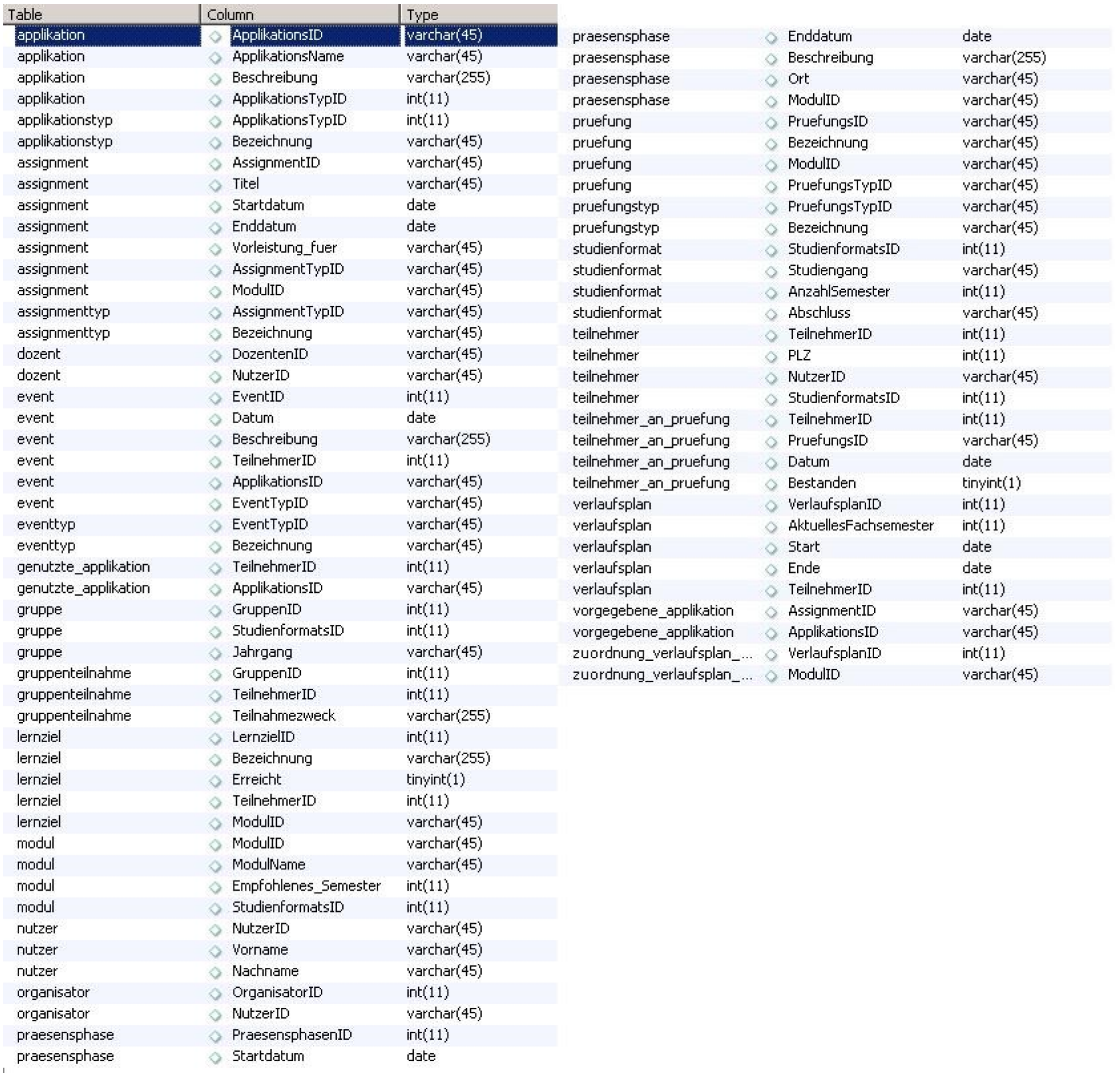

Table	Column	Type
applikation	ApplikationsID	varchar(45)
applikation	ApplikationsName	varchar(45)
applikation	Beschreibung	varchar(255)
applikation	ApplikationsTypID	int(11)
applikationstyp	ApplikationsTypID	int(11)
applikationstyp	Bezeichnung	varchar(45)
assignment	AssignmentID	varchar(45)
assignment	Titel	varchar(45)
assignment	Startdatum	date
assignment	Enddatum	date
assignment	Vorleistung_fuer	varchar(45)
assignment	AssignmentTypID	varchar(45)
assignment	ModulID	varchar(45)
assignmenttyp	AssignmentTypID	varchar(45)
assignmenttyp	Bezeichnung	varchar(45)
dozent	DozentenID	varchar(45)
dozent	NutzerID	varchar(45)
event	EventID	int(11)
event	Datum	date
event	Beschreibung	varchar(255)
event	TeilnehmerID	int(11)
event	ApplikationsID	varchar(45)
event	EventTypID	varchar(45)
eventtyp	EventTypID	varchar(45)
eventtyp	Bezeichnung	varchar(45)
genutzte_applikation	TeilnehmerID	int(11)
genutzte_applikation	ApplikationsID	varchar(45)
gruppe	GruppenID	int(11)
gruppe	StudienformatsID	int(11)
gruppe	Jahrgang	varchar(45)
gruppenteilnahme	GruppenID	int(11)
gruppenteilnahme	TeilnehmerID	int(11)
gruppenteilnahme	Teilnahmezweck	varchar(255)
lernziel	LernzielID	int(11)
lernziel	Bezeichnung	varchar(255)
lernziel	Erreicht	tinyint(1)
lernziel	TeilnehmerID	int(11)
lernziel	ModulID	varchar(45)
modul	ModulID	varchar(45)
modul	ModulName	varchar(45)
modul	Empfohlenes_Semester	int(11)
modul	StudienformatsID	int(11)
nutzer	NutzerID	varchar(45)
nutzer	Vorname	varchar(45)
nutzer	Nachname	varchar(45)
organisator	OrganisatorID	int(11)
organisator	NutzerID	varchar(45)
praesensphase	PraesensphasenID	int(11)
praesensphase	Startdatum	date
praesensphase	Enddatum	date
praesensphase	Beschreibung	varchar(255)
praesensphase	Ort	varchar(45)
praesensphase	ModulID	varchar(45)
pruefung	PruefungsID	varchar(45)
pruefung	Bezeichnung	varchar(45)
pruefung	ModulID	varchar(45)
pruefung	PruefungsTypID	varchar(45)
pruefungstyp	PruefungsTypID	varchar(45)
pruefungstyp	Bezeichnung	varchar(45)
studienformat	StudienformatsID	int(11)
studienformat	Studiengang	varchar(45)
studienformat	AnzahlSemester	int(11)
studienformat	Abschluss	varchar(45)
teilnehmer	TeilnehmerID	int(11)
teilnehmer	PLZ	int(11)
teilnehmer	NutzerID	varchar(45)
teilnehmer	StudienformatsID	int(11)
teilnehmer_an_pruefung	TeilnehmerID	int(11)
teilnehmer_an_pruefung	PruefungsID	varchar(45)
teilnehmer_an_pruefung	Datum	date
teilnehmer_an_pruefung	Bestanden	tinyint(1)
verlaufsplan	VerlaufsplanID	int(11)
verlaufsplan	AktuellesFachsemester	int(11)
verlaufsplan	Start	date
verlaufsplan	Ende	date
verlaufsplan	TeilnehmerID	int(11)
vorgegebene_applikation	AssignmentID	varchar(45)
vorgegebene_applikation	ApplikationsID	varchar(45)
zuordnung_verlaufsplan_...	VerlaufsplanID	int(11)
zuordnung_verlaufsplan_...	ModulID	varchar(45)

Abbildung 4.5: Datenbankschema zum Speichern der Nutzeraktivitäten von [Ack14]

Die Abbildung 4.5 bietet eine Übersicht über die Datenstruktur der Arbeit von [Ack14]. Diese soll den für den nachfolgenden Teil des Konzeptes kurz untersucht und erläutert werden. Die genauen Zusammenhänge, inklusive ausführlicher Erläuterungen und Abbildungen, sind in der Arbeit von [Ack14] zu finden.
Die Tabelle `applikation` enthält Daten zu den im Portal vorhandenen Portlets, inklusive eines Typs der in `applikationstyp` eingetragen ist.

Jedwede Art von Aufgaben (Hausaufgaben, Semesterarbeiten, etc.) werden in der Tabelle `assignments` abgelegt. Der zugehörige Typ des assignments wird in `assignmenttyp` festgelegt. Ebenso können Gruppen (`gruppe`) erstellt werden, dies könnte mit der '"Liferay"-eigenen Datenstruktur `usergroups` kombiniert werden. Events können in der Tabelle `event` mit der Verknüpfung zu `eventtyp` effizient abgespeichert werden. Ebenso können Lernziele (`lernziel`) vergeben werden. Die Datenstruktur verfügt bereits über die Möglichkeit Prüfungen (`pruefung` mit `pruefungstyp`) mit Vorlesungen, allgemeiner Modulen (`modul`) zu verknüpfen. Auch können Studienformate (`studienformat`), Teilnahmer an Prüfungen (`teilnehmer_an_pruefung`) und Verlaufspläne (`verlaufsplan`) abgespeichert werden.

4.3.3 Hooks

An dieser Stelle soll spezifisch auf die Thematik der Hooks in LifeRay eingegangen werden, da diese im vorangegangenen Abschnitt bereits mehrfach erwähnt wurden. Hooks sind eine wichtige Komponente, wenn es darum geht, Informationen über die Nutzung des Portals bekommen. Jede Nutzeraktivität in LifeRay führt zu sogenannten "Events", diese lösen dann sogenannte "Actions" aus, welche dann von einem Hook wahrgenommen werden können, was dann eine entsprechende Aktion im Hook auslöst.
Kurz zusammengefasst sind Hooks in Liferay Erweiterung von Portlets, mit deren Hilfe Interaktionen mit Selbigen beobachtet werden können. Hooks können in LifeRay so definiert werden, dass sie jede beliebige Interaktion mit einem Portlet erfassen können, sei eine Eingabe, beispielsweise in die Suchzeile des "Request" oder auch "Wegtam-Portlets" oder aber das simple Hinzufügen, Minimieren, Maximieren oder Entfernen eines Portlets. Liferay macht es dabei Entwicklern von Hooks relativ einfach, diese Interaktionen zu erfassen, da standardmäßig drei Arten von Actions zu Verfügung stehen (SimpleAction, Action, SessionAction), welche eine standartisierte Schnittstelle besitzen. Hierbei sei noch erwähnt dass es nicht notwendig ist, für jedes Portlet ein spezifischen Hook zu schreiben, denn Hooks können portalweit eingesetzt werden um Interaktionen zu überwachen, d.h. es kann beispielsweise ein Hook die Eingaben in Suchmaschinen überwachen, ein anderer Hook kümmert sich um Portletaktionen, ein dritter Hook beobachtet das Verhalten von Nutzern in Foren oder ähnlichem. Wie an diesen Beispielen zu sehen ist, sind Hooks universell im gesamten Portal einsetzbar, um Interaktionen zu beobachten.

Im Rahmen des Konzeptes soll nun darauf eingegangen werden, woher die für ein Empfehlungssystem notwendigen Daten kommen sollen. Liferay selbst bietet dafür in der sogenannten “Enterprise Edition“ eine Funktion namens “Liferay Audit“ an. Dies ist ein Service, der über vorgefertigte Hooks und Datenstrukturen in der Lage ist, Interaktionen mit dem Portal entsprechend festzustellen und für eine spätere Verarbeitung abzuspeichern. Da im momentanen Zustand des Portals nur “Community Edition“ von Liferay vorliegt, ist diese Audit-Komponente leider nicht Bestandteil dieser Version. Es ist aber möglich, da die Grundstruktur (Liferay “MessageBus“) für den Liferay Audit-Service in der “Community Edition“ trotzdem vorhanden ist, diese Ansammlung von Hooks für Portlets mit einem gewissen Aufwand zu ersetzen, in dem sie selbst implementiert werden. Hierbei soll nicht verschwiegen werden das dies nur mit einem Aufwand möglich ist, da die Hooks und die zugehörigen Datenstrukturen für die Speicherung selbst konzipiert und implementiert werden müssen. Dies ist notwendig, um eventuelle Inkompatibilitäten in zukünftigen Versionen von LifeRay zu verhindern.
Die Arbeit von [Ack14] zeigte bereits eine brauchbare Version einer Beobachtungskomponente für die Zusammenarbeit mit einem Empfehlungssystem. Dabei wurde, um die Nutzeraktivitäten festzuhalten und auszuwerten das Datenbankschema in Abbildung 4.5 erstellt. Dieses Schema bietet einen hervorragenden Ansatzpunkt für weitere Arbeiten am Portal und an der Beobachtungs- bzw. Empfehlungskomponente. Für Hooks ist dabei die Tabelle `event` in Verbindung mit `eventtyp` sehr wichtig.
Auch sollten, um das Empfehlungssystem performant zu gestalten, nicht immer Empfehlungen nach jeder Änderung neu errechnet werden. Hierzu ist in erster Linie wichtig, dass die Möglichkeit geschaffen wird, Veränderungen in den für das Empfehlungssystem relevanten Tabellen der Datenbank überhaupt zu erfassen. Dieses Erfassen sollte im Rahmen eines “Observer-Patterns“ geschehen, welches nur die für das System relevanten Änderungen überwacht und dann entsprechend weitermeldet. Auch ist es wichtig, dass einige Empfehlungen, wie beispielsweise bei der Suche, auf Vorrat berechnet werden können. Hierzu könnten beispielsweise Suchbegriffe so indexiert werden, dass ähnliche Begriffe ebenfalls vermerkt werden, und so die Berechnung nur zu Beginn aufwändig wäre.
Alle relevanten Änderungen sollten ebenfalls priorisiert werden, besonders wenn ersichtlich ist, dass die Folgen dieser Änderungen zu einem Berechnungsstau führen würden. Hierzu wäre es sinnvoll, die Veränderungen in Gruppen einzuteilen und diese dann gesammelt als Transaktion an die Datenbank zu schicken.

Dies sollte am besten in Zeiten geringer Aktivitäten am Portal geschehen, z.B. im Intervall der Wartungsarbeiten. Lediglich wichtige Änderungen sollten sofort, aber mit minimaler Last ,auf dem Empfehlungssystem ausgeführt werden. Änderungen durch Nutzer sollten, wie von [Ack14] empfohlen erst nach dem Log-Out des Nutzers erfolgen, um den Arbeitsfluss im Portal nicht zu stören.

4.3.4 Identifizierte items aus Abschnitt 4.2

Bezug nehmend auf die in Abschnitt 4.2 genannten **items**, wird an dieser Stelle nun genauer auf die notwendigen bzw. schon vorhandenen Datenstrukturen eingegangen, die für eine effiziente Verwaltung dieser **items** notwendig sind. Hierbei werden nur die für ein Empfehlungssystem relevanten Attribute genauer betrachtet, andere Attribute wie beispielsweise ein *Erstellungsdatum* sind nicht zwangsläufig wichtig und notwendig für ein Empfehlungssystem, weswegen sie an dieser Stelle der Betrachtung außen vor bleiben.

Prüfung

Die Datenstruktur für die Prüfungen existiert schon in der Verwaltung der Universität Rostock, nur fehlt es in der Datenbank des Portals an der notwendigen Datenstruktur. Allerdings wurde bereits durch die Arbeit von [Ack14] eine zusätzliche Datenbank eingeführt, welche eine Struktur für Prüfungen enthält. Diese Struktur sollte erweitert werden, da soweit alle notwendigen Attribute für eine Prüfung dort vorhanden sind.

Literatur

Für die Literatur existiert ebenfalls keine Datenstruktur in der Standardausführung des Portals, ebensowenig in der Datenbank von [Ack14]. Deswegen ist es notwendig eine solche Struktur herzustellen. Die Datenstruktur sollte folgende Attribute enthalten: *ID, Titel, Autor, Abstract, ?UB-Code?, ?Verfügbarkeit in der UB?*. Besonders der *Abstract* ist wichtig, da über ihn schnelle Ergebnisse im inhaltsbasierten Ansatz erzielt werden können und so vorerst der zeitaufwändige Volltextvergleich vermieden werden kann. Natürlich sollte die Volltextindizierung vollzogen werden, um noch feinere Abstufungen in Ähnlichkeiten von Texten feststellen zu können, dazu müssen diese in digitaler Form vorliegen, weswegen der Vergleich über den *Abstract* für den Beginn die bessere Variante darstellt.

Zudem kann jeder Nutzer nach der Empfehlung über das Lesen des *Abstracts* für sich feststellen, ob dieser Text für ihn passend sein könnte. Die *?Verfügbarkeit?* in der Universität-Bibliothek wäre ein optionales Attribut, aber dennoch sehr sinnvoll um dem Nutzer im Zuge der Empfehlung die Möglichkeit zu geben, wenn diese genau trifft, sich das entsprechende Werk, so vorhanden, aus der Bibliothek ausleihen zu können. Zudem wäre es über eine Verknüpfung des Ausleihbestandes der Bibliothek mit den abgespeicherten Empfehlungen möglich zu erfassen, ob eine Empfehlung genau war oder nicht, dazu näheres in Abschnitt 4.7.

Hausaufgaben

Dieser Abschnitt betrifft Hausaufgaben wie auch Semesterarbeiten. Die in der Arbeit von [Ack14] entworfene Datenstruktur kann auch für diese **items** genutzt werden (`assigment`). Dort sind bereits viele wichtige Attribute enthalten inklusive der Verknüpfung zu einer Vorlesung, hier durch eine *ModulID*. Ebenso ist es möglich einen *Typ* festzulegen. Nicht enthalten ist die Möglichkeit, eine Aufgabe zu *priorisieren* (z.B. für eine Prüfung), dies sollte nachgetragen werden, um so solche Aufgaben besser empfehlen zu können. Die Empfehlung kann hier über den hybriden Ansatz mit Fokus auf den Kollaborativen geschehen.

Vorlesungen

Für Vorlesungen allgemein existiert keine spezifische Datenstruktur in "Liferay", dafür existieren in der Datenstruktur von [Ack14] rudimentäre Ansätze. Hier werden nicht explizit Vorlesungen abgespeichert, sondern Module. Dies könnte aber für zukünftige Verknüpfungen mit Vorlesungen an sich genutzt werden. Hierzu wäre es notwendig, eine Datenstruktur zu erstellen, die eine *ID*, die *Art der Vorlesung* (also regulär, zusätzlich, etc. siehe Abschnitt 4.2), den *Namen*, den *Dozenten* und den *Inhalt* enthält. Der *Inhalt* könnte erfolgsversprechend vom inhaltsbasierten Ansatz genutzt werden, der kollaborative könnte über Bewertungen von Vorlesungen bzw. über eingeschriebene Studenten Empfehlungen geben.

Übungstermine/-gruppen

Übungen sind wie die Vorlesungen an ein Modul gebunden. Dies sollte für eine entsprechend Struktur genutzt werden, um so das LSF für ein Empfehlungssystem nutzbar abzubilden. **Übungen** benötigen wie die **Vorlesungen** auch eine *ID*, einen *Name*, einen *Inhalt*, eine *Zuordnung* zu einer Vorlesung (dies sollte über die *ModulID*) geschehen, einen *Ort*, eine *Zeit* und eventuell eine *?Größe?*, um abschätzen zu können, wie vielen Leuten diese **Übung** empfohlen werden kann oder ob noch Platz vorhanden ist.
Ebenso sollte eine Struktur für Übungsgruppen entstehen, "Liferay" bietet als Grundfunktion die Erstellung von Gruppen an, dies sollte entsprechend genutzt werden.

Lerngruppen/-partner

Ähnlich verhält es sich mit den **Lerngruppen** bzw. **Lernpartnern**, auch hier sollte die bestehende Datenstruktur für Gruppen `usergroup` genutzt werden. Allerdings sollte ebenso überprüft werden, inwieweit die Gruppentabelle von [Ack14] genutzt werden kann. Empfehlungen für **Lernpartner** oder **Lerngruppen** könnten über Freundeslisten oder über jene Gruppen entstehen. Bei den Gruppen selbst wäre es sinnvoll, noch ein Attribut für einen "Offenheits-Status" einzuführen, um so bewusst Gruppen für Empfehlungen priorisieren zu können.

Dateien

Für Dateien ist in der Arbeit [Wei14] und im "Google Drive Quick Links" Portlet eine simple aber für die Zukunft verwendbare Datenstruktur erzeugt worden. Diese enthält zwar "nur" Verknüpfungen von **Nutzern** mit **Dateien** über eine *URL*, dennoch kann diese Struktur als Grundlage für ein Weiterentwicklung genutzt werden. In Abschnitt 4.4 wird die Idee eines Cloud-Dienstes aufgegriffen und diskutiert werden, dies könnte bei einem portaleigenen Cloud-Dienst zu gut beobacht- und nutzbaren Nutzeraktivitäten führen, welche wiederum von einem Empfehlungssystem genutzt werden können. Zudem würden Nutzer so nicht ausgeschlossen werden, wenn sie keine Zugang zu dem Cloud-Dienst haben.
Eine Datenstruktur für Dateien sollte folgende Attribute enthalten: *ID, Speicherort, ?Speichermedium?, ?Gruppe?*. Unter dem *Speicherort* ist der genaue Ablageort (URL) einer Datei inklusive deren Zugangsbeschränkungen (z.B. durch Cloud-Dienste) zu verstehen.

Das *Speichermedium* ist nicht primär wichtig, dies wäre nur interessant, wenn es sich um ein temporäres Speichermedium handeln würde. Die *Gruppe* gehört ebenfalls zu den optionalen Attributen, dennoch wäre es sinnvoll dieses umzusetzen, da hierüber **Dateien** gruppiert oder gar von der Empfehlung ausgeschlossen werden könnten, was dem Empfehlungssystem durch die Aggregation zu Gruppen die Berechnung vereinfachen würde.

Websites

Für **Websites** bietet "Liferay" bereits in seiner Grund-Installation eine Datenstruktur an. Ob diese allerdings auch anderweitig z.B. von Portlets genutzt wird, konnte nicht eindeutig bestimmt werden. Deswegen wird empfohlen vorsichtshalber eine andere Tabelle mit der gleichen Struktur anzulegen, um die Aufrufe von **Websites** für ein Empfehlungssystem nutzbar erfassen zu können. Dies sollte durch einen Hook geschehen. Eine weitere Attribut-Ergänzung wäre die optionale Aufnahme von *?Gruppen?* für **Websites**, da diese dann entsprechend aggregiert werden könnten.
Bei den **Websites** könnte der kollaborative Ansatz am erfolgversprechensten sein, für die inhaltbasierte Komponente müsste der Inhalt analysiert werden. Da dies aber zu einer großen Datenmenge führen dürfte, wäre es sinnvoller, als Erstes den kollaborativen Ansatz umzusetzen und den Inhaltsbasierten auf Leistungsfähigkeit und Datenmenge zu überprüfen.

Sonstige Veranstaltungen im Semester

Für die sonstigen Veranstaltungen war ebenfalls keine brauchbare Datenstruktur zu finden. Hier sollte ebenfalls eine effiziente Datenstruktur erstellt werden, welche die folgenden Attribute umfasst: *ID, Name, Ort, Beginn, Ende, Inhalt*. Über den zeitlichen Bezug kann die Empfehlungsgebung entsprechend stattfinden, der *Inhalt* erlaubt eine Ähnlichkeitsberechnung zu den bereits besuchten Vorlesungen/Modulen. Ebenso können über den kollaborativen Ansatz Ergebnisse erzielt werden, wenn beispielsweise die Aktivierung von Links durch Hooks erfasst werden kann.
Die notwendigen Daten sollten in der Verwaltung vorliegen und entsprechend eingepflegt werden, andernfalls können diese auch aus Mails extrahiert werden.

Suchen

Mit der **Suche** kann ähnlich verfahren werden, wie mit den **Websites**. Da es hier weder in "Liferay" noch in der Datenbank von [Ack14] eine nutzbare Datenstruktur gibt, muss eine solche erstellt werden. Dies sollte keine große Herausforderung darstellen, da hier lediglich *Suchbegriffe* erfasst werden müssen, welche mit dem inhaltsbasierten Ansatz verglichen und gruppiert werden können. Auch sollte es möglich, sein in der Datenstruktur die *Häufigkeit* bei der Sucheingabe festzustellen. Schwieriger wird die Verknüpfung dieser Struktur mit dem Suchen im '"Request"-Portlet. Hier sollte wie bereits erwähnt eine Gruppierung der vorhandenen Suchbegriffe im Voraus geschaffen werden, damit die Last für eine Berechnung direkt bei der Eingabe entweder ganz oder zu einem großen Teil abgefangen wird.

Räume/Raumpläne

Räume sowie die zugehörigen **Raumpläne** gehören auch zu den **items**, für welche eine eigene Datenstruktur erstellt werden muss. Dabei sollte die Daten an sich in der Verwaltung der Fakultäten zu finden und einfach übertragbar sein. Wichtig für ein Empfehlungssystem sind bei den **Räumen**: *ID, Raumnummer, Adresse (Gebäude), Verfügbarkeit, Raumplan*, wobei sich die *Verfügbarkeit* entweder aus dem Raumplan ergibt (dazu muss dieser zeitlich überwacht werden) oder durch das Feststellen von Aktivitäten im Raum. Die **Raumpläne** sollten folgende Attribute enthalten: *Uhrzeit_Beginn, Uhrzeit_Ende, ?Thema der Veranstaltung?.*
Räume können durch Auslösen des Nutzer manuell empfohlen werden. Ebenso ist es denkbar durch das Feststellen des aktuellen Nutzerstandortes (durch WLAN-Router) **Räume** in der Nähe zu empfehlen.

Gastronomische Einrichtungen

Die **gastronomischen Einrichtungen** stellen eine weitere Art von **items** dar, für diese gibt ebenfalls keine nutzbaren vorhandenen Datenstrukturen. Deswegen sollte eine kleine Datenstruktur erstellt werden, welche die folgende Attribute aufweist: *ID, Name, Art, Adresse, Öffnungszeiten, Bewertung_allgemein, Speiseplan.* In so einer Datenstruktur können dann alle gastronomischen Einrichtungen erfasst werden.

Das Attribut *Art* steht hierbei für die unterschiedlichen Ausrichtungen von solchen Einrichtungen, also beispielsweise “Mensa“, “Kneipe/Bar“ oder “Lieferdienst“, *Speisepläne* umfassen das gesamte Angebot, deren Preise und könnten gleichzeitig auch Verknüpfungen für Bewertungen von Speisen bieten.
Empfehlungen sollten über den hybriden Ansatz geschehen, um die Genauigkeit der Empfehlung zu erhöhen.

4.3.5 Zusammenfassung

Zusammenfassend lässt sich zu der technischen Ebene des Konzeptes sagen, dass die Erarbeitung des Konzeptes zu den schwierigeren, wenn nicht schwierigsten Aufgaben dieser Arbeit zählt. “Liferay“ bringt zwar viele Datenstrukturen und Portlets in seiner Grundinstallation mit, allerdings sind davon relativ wenige auch für Verwendung eines Empfehlungssystems ausgelegt. Hier herrscht eindeutiger Nachholbedarf seitens “Liferay“. Bisher kann dieser Mangel an nutzbaren Strukturen nur durch die Arbeit entweder im Rahmen von studentischer Hilfsarbeit, von Projekten oder Abschlussarbeiten Stück für Stück beseitigt werden. Es wäre schön, würde “Liferay“ in Zukunft entweder direkt ein System in seine Portal-Software integrieren oder aber wenigstens mehr Datenstrukturen haben, die Anknüpfungspunkt für Empfehlungssysteme bilden. Die beiden bedeutensten für ein Empfehlungssystem verwendbare Strukturen sind die Tabellen `ratingsentry` (in der portalweit Bewertungen gesammelt werden), `usergroups` (in der Gruppen für die Portalnutzer erfasst werden), sowie alle Tabellen aus der Datenbank von [Ack14]. Dies ist positiv, da diese Tabellen schon für den hybriden Ansatz allgemein, und selektiv für den inhaltsbasierten oder kollaborativen Ansatz, genutzt werden können und das alle Daten zentral gesammelt und auch so verwaltet werden können. Nun wäre es natürlich möglich, das im Falle einer Umstellung des gesamten universitären Betriebs auf das Portal, durch die hohe Nutzerzahl, welche dann durch das Projekt “KOSMOS“ zusätzlich entstehen wird, die Tabelle `ratingsentry` sehr schnell sehr groß werden kann. Es wäre daher bei zukünftige Entscheidungen wichtig zu evaluieren, ob die Datenmenge noch gut in einer Tabelle handhabbar ist oder ob begonnen werden sollte, diese Tabelle beispielsweise aufzuteilen, um so alle Bewertungen halten zu können.

Um das Empfehlungssystem an sich performant zu gestalten, wird es als wichtig erachtet, dass Operationen, die nur mit Empfehlungsberechnung zu tun haben, möglichst mit Referenzen und Indizes arbeiten, um so die Datenlast und den Datenverkehr so gering wie möglich zu halten. Das Empfehlungssystem von [Ihl14] zeigt bereits, dass dies möglich ist und auch in einer Version ohne bedeutende Optimierung zu schnell berechneten Ergebnissen führen kann. Für das Endergebnis und die Präsentation des Selbigen müssen diese Referenzen und Indizes am Ende natürlich wieder in eine für Menschen, sprich den Nutzer, verarbeitbare Darstellung überführt werden. Auch deswegen ist es wichtig, möglichst viele Informationen für das Empfehlungssystem gut erreichbar zu speichern. Ebenso wäre es sinnvoll, Empfehlungen lediglich neu zu berechnen, wenn eine entsprechende Änderung in der Datenstruktur auftritt. Dies kann recht einfach durch ein "Observer-Pattern" gelingen, dafür könnte man in "Liferay" einen Hook einsetzen. Diese Änderungen sollten ebenfalls der Performanz wegen priorisiert und zu Transaktionen zusammengefasst werden.

Generell wird empfohlen, aufgrund der zu erwartenden Datenmenge und der damit einhergehenden Last im Verlaufe des Betriebs des Portals zu evaluieren, ob es nicht sinnvoll ist, für das Empfehlungssystem eine eigene Datenbank zu entwerfen und eine eigene Arbeitsmaschine aufzusetzen, um so die Performanz auf einem vernünftigen, nutzbaren Niveau zu halten.

4.4 Wünschenswerte Erweiterungen

Dieser Teil des Konzeptes widmet sich den Erweiterungen des Portals, die in Zukunft notwendig oder wünschenswert wären, um zusätzliche Empfehlungen generieren zu können oder die Ergebnisse der Berechnung verbessern zu können.

4.4.1 Gruppenfunktion

Eine Komponente, die es dem hinter *myKOSMOS* stehenden Empfehlungssystem erleichtern würde, Empfehlungen zu generieren, wäre eine Gruppenfunktion direkt in *LifeRay*. Nach den bisherigen Erkenntnissen beim Umgang mit dem Portal, ist eine solche Komponente noch nicht implementiert oder integriert.

Allerdings bietet *Liferay* eine solche Komponente, teilweise mit einem vergleichbaren Umfang an Funktionen, an. Sie trägt die Bezeichnung "Members Portlet" und ist Teil der "Social Networking Tools" [lif15e]. Nach der Beschreibung im Wiki von *LifeRay* ist das "Members Portlet" ein Portlet, welches die Mitglieder einer Gemeinschaft oder Organisation, zu der das Portlet hinzugefügt wurde, in Selbigem als Liste darstellt, als schnelle Möglichkeit, um auf die persönliche Seite eines jeden Nutzers zugreifen zu können [lif15c]. Dieses Portlet könnte als Grundlage für eine zukünftige Entwicklung eines "Gruppen-Portlets" genutzt werden. Dies müsste genauer untersucht werden. Eine solche Komponente würde generell die Gruppierungen vereinfachen, und ebenso damit den restlichen Ablauf der Empfehlungsberechnung. Denkbar als zusätzliche Komponente für Gruppenerstellung wäre die direkte Übernahme von bestehenden Gruppen aus sozialen Netzwerken, wie z.B. Facebook. Hierbei bestünde natürlich wieder die Schwierigkeit, die Nutzer von "myKOSMOS" mit denen aus "Facebook" zu verknüpfen.
Natürlich muss für diese Gruppenfunktion auch eine geeignete Datenstruktur erstellt werden, um die **Gruppen** effizient und sinnvoll verwalten zu können. Diese Datenstruktur sollte aus Sicht eines RS eine *Gruppen-ID*, den *Gruppennamen* und die *Gruppenart* (Lerngruppe für Hausaufgaben, Prüfungen oder ähnliches) enthalten. Weitere wichtige Daten sind natürlich die Nutzer, die *Mitglieder* dieser Gruppe sind, damit verbunden auch die *Größe* dieser Gruppe, und der *Studiengang* der Gruppenmitglieder. Denkbar wäre ebenso eine Erweiterung der Informationen zur Gruppe um einen *Status*, der angibt, ob diese Gruppe noch Mitglieder aufnehmen will oder nicht. Aus diesen Informationen lässt sich bereits für die Gruppe eine bessere Empfehlung erstellen, ausgeweitet auf alle anderen Nutzer lassen sich z.B. über die Studiengänge Gemeinsamkeiten finden. Ebenso kann hier der kollaborative Ansatz, wie auch der hybride Ansatz gute Ergebnisse erzielen, da die Gruppen schon von Nutzern festgelegt worden sind, und so bereits ein Teil der Aggregationsberechnungen entfällt.

"Liferay" besitzt generell die Möglichkeit, Gruppen anzulegen, allerdings nur von Seiten der Administratoren und versteckt vor den normalen Nutzern des Portals [lif15b]. Es muss die Möglichkeit überprüft werden, ob solche Privilegien wie Gruppengründung, in eingeschränktem Maße mit Veränderungen, den normalen Nutzern zugänglich gemacht wird oder ob es sinnvoller ist, diese Funktion, wie oben beschrieben in ein eigenes Portlet zu verpacken.

Eine ähnliche Funktion bietet das in Abschnitt 2.4 beschriebene “Skype Chat & IM“-Portlet an, allerdings ist die Gruppenkomponente auf Skype beschränkt und setzt demzufolge natürlich einen bestehenden Skype-Account voraus. Im Sinne der Integration des Nutzers in das Portal und um die Komplexität der Gruppenerstellung und -verwaltung so gering wie möglich zu halten, ist es sinnvoll, eine nahtlose Gruppenfunktion im Portal selber anzubieten und eventuelle Übernahmen aus sozialen Netzwerken oder von anderen Diensten wie Skype möglich zu machen. Dies sollte sinnvollerweise sowohl vorrangig in der Gruppe als auch für jeden Nutzer privat möglich sein.

4.4.2 Cloud-Dienst(e)

Eine weitere wünschenswerte Erweiterung des Portals wäre die Ergänzung um weitere Cloud-Dienste und/oder einen übergeordneten Dienst, der mit allen großen Cloud-Anbietern kompatibel ist, um die Verteilung von und den Zugang zu **Dateien** zu vereinfachen. Es wurde, wie bereits in Abschnitt 2.4 erwähnt, von [Wei14] ein Portlet konzipiert und implementiert, welches eine Verbindung zu “Google Drive“ schafft, um dort gespeicherte **Dateien** abrufen zu können. Natürlich ist dies ein erster Schritt in die richtige Richtung, da Cloud-Diensten eine immer größer werdende Bedeutung zuteil wird, wenn es um schnelle (die nötige Infratruktur vorausgesetzt) und einfache Verteilung von Dokumenten geht. Jedoch würde es den Arbeitsfluss im Portal selber signifikant verbessern, wenn in das Portal die Möglichkeit mit implementiert würde, Dokumente nicht nur über “Google Drive“ zu teilen, sondern wahlweise andere Dienste mit einzubinden und/oder im Portal direkt einen Cloud Dienst anzubieten. Der Vorteil der ersten Variante wäre natürlich, dass jede Art von Speicherung und allem was damit zusammenhängt, an externe Anbieter ausgelagert werden würde. Dies würde sonst enorme Kosten generieren, würde man den Cloud-Dienst selber anbieten. Zusätzlich würde sich auch der Aufwand für eine Integration und Implementation in das Portal bei dem Anbieten externe Dienstleistungen im Cloud-Bereich in Grenzen halten, da nur ein Interface, sprich Portlet, für die Nutzung dieser Dienste im Portal geschaffen werden müsste. Für das RCS bedeutet dies natürlich einen etwas erhöhten Aufwand, um an eventuelle Nutzungsdaten zu kommen, da diese, wenn überhaupt nur bei der Nutzung des Dienstes durch das Portal selber gesammelt werden können. Zudem würden Nutzer aktiv von einer Empfehlung dadurch ausgeschlossen, wenn sie beispielsweise keinen Zugang (Konto) zu einem externen Anbieter hätten.

Der Vorteil der Methode (s)einen eigenen Cloud-Dienst im Portal zur Verfügung zu stellen liegt darin, dass der Betreiber des Portals, also die Universität Rostock, die volle Kontrolle über die Nutzung des Portals und natürlich der Dateien in diesem Portal hat. Dies kann als positiv angesehen werden, da nun Datenschutz-Richtlinien selbst definiert, umgesetzt und auch überprüft werden können, ohne das man sich auf externe Dienstleister verlassen muss, meist ohne die Möglichkeit, auf so etwas wie Datenschutz und auch Datensicherheit einflussnehmen zu können. Das alles stellt natürlich einen relativ hohen Aufwand da, es gilt nun abzuwägen, was langfristig sinnvoller und im Rahmen des Budgets des Projektes *myKOSMOS* ein vertretbarer Aufwand für einen Cloud-Dienst ist. Es wäre ebenso denkbar, Cloud-Anbieter am Markt zu suchen und zu evaluieren (Datensicherheit/-schutz, Budget), um so eventuell einen geeigneten Dienstleister für einen Cloud-Dienst im Portal zu finden, da die meisten Cloud-Dienste auch sogenannten Business-Plans anbieten, also Angebote für Firmen, die großen Speicherbedarf haben, aber sich eine eigene Cloud nicht leisten können oder wollen. Natürlich greift hier wieder das oben beschriebene Problem mit der schwierigen Überwachung der Nutzung der Daten und des Umganges mit selbigen. In den Zeiten der Spähprogramme von Geheimdiensten, der Wirtschaftsspionage und der Möglichkeit, dass sich potentiell in jedem Programm (Stichwort Cloud) eine sogenannte Backdoor, also ein geheimer Zugang für Dritte (Geheimdienste, Hersteller, etc.) befindet, sind gute Absicherungen gegen unbefugten Zugang und Nutzung der Daten mehr als notwendig, quasi eine Grundvoraussetzung.
Liferay bietet seit kurzer Zeit einen hauseigenen Cloud-Dienst an. Dieser trägt den passenden Namen *LifeRay Sync* [lif15a]. Nun gilt es unter den oben bereits genannten Gesichtspunkten zu evaluieren, ob *Liferay Sync* eine sinnvolle und nutzbare Alternative zu anderen Cloud-Diensten wie Copy, Google Drive, Dropbox, etc. darstellt.

4.4.3 Suche

Mit dieser Erweiterung ist nicht prinzipiell die Implementation einer Suche an sich gemeint, sondern die Erweiterung der bereits in Abschnitt 2.4 beschriebenen "Wegtam-Suche" um Google-ähnliche Anpassungen der Suche bzw. des Suchergebnisses. Es ist bereits möglich, die Suche in Wegtam durch die Nutzung von Suchprofilen vor der eigentlichen Suche oder durch Filter auf das Suchergebnis, selbiges anzupassen.

Dies stellen bereits sehr gute Grundfunktionalitäten da, welche durch ein Empfehlungssystem erweitert werden können. Dies soll so geschehen wie ein paar Seiten vorher im Abschnitt “Suche“ beschrieben. Auf der einen Seite wäre es sinnvoll, wie beispielsweise “Google“ oder “DuckDuckGo“ es bereits umgesetzt haben, bereits bei der Eingabe von Suchbegriffen passende Ergänzungen oder gar Alternativen vorzuschlagen. Auf der anderen Seite könnte man dem Nutzer entweder gleich eine Seite vorschlagen, sei es nun der Index der Seite oder eine spezifische Unterseite einer Website, oder aber ihm ein Suchprofil bzw. Filter empfehlen. Weiterführend könnte man Nutzern gleich ein solches Profil als Standardeinstellung setzen, ebenso wäre dies mit Filtern denkbar, um so dem Nutzer ein auf ihn zugeschnittenes Suchergebnis zu präsentieren.
Dazu ist es allerdings nötig, das effiziente Datenstrukturen angelegt werden, in denen **Suchen** mit ihrem *Suchbegriffen* und *Suchergebnissen* vernünftig abgespeichert werden können.

4.4.4 Gastronomie

Für diese Erweiterung der Empfehlungsgebung kann man den Vergleich zu einigen bekannten Apps für den universitären Alltag ziehen, z.B. die ’"CampusApp“ oder “Mensa“. Diese Apps bieten entweder als Grundfunktion (“Mensa“) oder als eine von vielen Funktionen “CampusApp“ die Möglichkeit an, sich die Mensen der jeweiligen Universitäten inklusive deren *Speiseplan* für einen spezifischen Tag, meist einen aus einer ganzen Woche, anzusehen. Diese Funktion könnte im Rahmen eines weiteren Portlet oder Gadget für das Portal “myKOSMOS“ umgesetzt werden. Weiterhin könnten die Angaben um die *Preise*, wie sie beispielsweise in den Mensen der Universität am Eingang zu finden ist, ergänzt werden. Bewertungsstrukturen (*Bewertungen*) sind in jeder App zu finden, weswegen es auch angezeigt ist, diese Grundfunktionen mit in ein etwaiges Portlet zu übernehmen. All dies ließe sich auch für anderweitige gastronomische Einrichtungen übertragen.
Wichtig ist, dass Empfehlungen wie in Abschnitt 4.2 beschrieben, gegeben werden. Es ist sinnvoller, erst nur die gastronomische Einrichtung an sich zu empfehlen und dann bei einem ausreichend differenzierten Nutzerprofil konkrete Gerichte vorzuschlagen. Ebenso sollte die Preiskategorie eines jeden Nutzers mit berücksichtigt werden. Berechnung können hier am besten über den hybriden Ansatz geschehen.

4.4.5 Shop/Store

Diese Erweiterung wird als Erweiterung der bereits in “Liferay“ vorhandenen Struktur eines Online-Shops konzipiert. Hierbei geht es darum, eine Plattform für beispielsweise nicht mehr fürs das Studium benötigte Bücher bereitzustellen, um so einen zentralen Anlaufpunkt zu schaffen, als Ergänzung der schwarzen Bretter in den einzelnen Fakultäten sowie den Mensen, etc.. Ebenso wäre es denkbar, den Bestand an Büchern der Universitätsbibliothek mit in diese Plattform zu integrieren. Dies sollte an geeigneter Stelle genauer evaluiert werden. Zusätzlich könnte diese Plattform z.B. für den Verkauf der Kapuzenpullover und -jacken des IEF-FSR, also kurz für fakultätsbezogene Kleidung genutzt werden, dies dürfte den Zulauf erhöhen, da nun die Angebote dauerhaft und präsenter verfügbar sind und nicht nur beispielsweise in einer E-Mail erwähnt werden.
Die Erweiterung um einen Shop/Store des Portals würde neue **items** in die Masse der empfehlungswürdigen **items** mit einbringen, gleichzeitig würde damit auch eine größere Basis für die Berechnungen des Empfehlungssystems geschaffen werden. Empfehlungen sollten nach der hybriden Methode berechnet werden, um die Präzision zu erhöhen und könnten in dem Shop-Portlet auf einer Startseite platziert werden.

4.5 Datenschutz

In diesem Teil des Konzeptes soll der Datenschutz und die Umsetzung des Selbigen im Vordergrund stehen. Dabei wird auf den Datenschutz generell eingegangen und besonderes Augenmerk auf das Thema der “k-Anonymität“ gelegt. Dies soll bei der zukünftigen Umsetzung des Konzeptes helfen, etwaige Risiken, wie sie beim Umgang mit sensiblen Daten entstehen können, zu minimieren.

Datenschutz ist ein wichtiges und sensibles Thema, gerade wenn es um den Umgang mit sensiblen Nutzerdaten geht, was sich bei einem Recommendation System kaum vermeiden lässt. Hier muss der Fokus darauf gerichtet sein, das die Daten bei der Erhebung und Weiterverarbeitung durch das Empfehlungssystem ausreichend geschützt sind, im Sinne der informationellen Selbstbestimmung.
Dies kann dadurch geschehen, das man einerseits das System nach außen absichert und den Teil des Systems, wo mit sensiblen und anderen Daten gearbeitet wird, besonders geschützt wird (Datensicherheit).

Andererseits ist es für das Empfehlungssystem viel wichtiger, die Daten für die Erhebung und Weiterverarbeitung so zu verändern und zu anonymisieren, das gelungene Angriffe auf das System keine brauchbaren Daten liefern würden (Datenschutz). Zudem bedeutet der Schutz der Computer nicht unbedingt den Schutz von sensiblen Daten [Swe02].
Der Begriff der Datensicherheit ist eng mit dem des Datenschutzes verknüpft, dennoch muss darauf geachtet werden, dass diese Begriffe nicht verwechselt werden. Die Datensicherheit beschäftigt sich mit der technischen Sicherheit und hat das Ziel Daten jedweder Art gegen Verlust, Manipulation/Verfälschung, Zerstörung oder andere Bedrohungen zu schützen. Deswegen muss neben dem Datenschutz auch ein Fokus auf die Datensicherheit gelegt werden, da die Datensicherheit eine Voraussetzung für effektiven Datenschutz ist.

Da ein Empfehlungssystem viel mit aggregierten Daten arbeitet, ist die Thematik der "k-Anonymität" von enormer Bedeutung, um die dabei zwangsweise mit erhobenen sensiblen Daten in der aggregierten Menge vor Unbefugten zu schützen; das bedeutet sie bis zur Unkenntlichkeit zu anonymisieren. Dieses Vorgehen im Rahmen der "k-Anonymität" wird im nun folgenden Abschnitt beschrieben, dabei wird auf die Umsetzung, die Vorteile und die Nachteile dieser Technik eingegangen.

k-Anonymität

Dem Thema der *k-Anonymität* soll hierbei besonderes Augenmerk gelten. Gerade ein Recommendation System arbeitet fast ausschließlich mit aggregierten Daten, weswegen diese Daten mit besonderer Vorsicht gehandhabt werden müssen. Eine solche Vorsichtsmaßnahme stellt dabei die sogenannte *k-Anonymität* dar.
Daten bieten k-Anonymität, falls die identifizierenden Informationen eines jeden einzelnen Individuums von mindestens k-1 anderen Individuen ununterscheidbar sind, und somit die eine korrekte Verknüpfung mit den zugehörigen sensiblen Attributen erschwert wird [Swe02]. Bei der *k-Anonymität* handelt es sich grundsätzlich um einen Kompromiss zwischen einem höheren Maß an Datenschutz und dem Verlust der Genauigkeit der Daten auf der anderen Seite. Zum besseren Verständnis des Problems wird nun ein Beispiel zur Veranschaulichung gegeben.

	Nicht-sensible Attribute			Sensibles Attribut
	Alter	**Nationalität**	**PLZ**	**Erkrankung**
1	27	Russisch	13152	Herzerkrankung
2	25	Tschechisch	13253	Herzerkrankung
3	23	Japanisch	13052	Virusinfektion
4	23	Amerikanisch	13141	Virusinfektion
5	51	Indisch	14752	Krebs
6	55	Russisch	14865	Herzerkrankung
7	47	Amerikanisch	14452	Virusinfektion
8	49	Deutsch	14422	Virusinfektion
9	33	Amerikanisch	13053	Krebs
10	37	Indisch	13053	Krebs
11	36	Japanisch	13254	Krebs
12	31	Amerikanisch	13153	Krebs

Tabelle 4.1: Nicht-anonymisierte Patientendatentabelle nach [MKGV07]

Im nächsten Schritt wird die Tabelle generalisiert, dass bedeutet das alle nicht-sensiblen Attribute verallgemeinert oder elemeniert werden. Im konkreten Beispiel bedeutet das, dass Altersgruppen gebildet werden, ebenso wird die PLZ verallgemeinert und die Nationalität der Patienten wird komplett aus dem Datensatz gestrichen.

	Nicht-sensible Attribute			Sensibles Attribut
	Alter	**Nationalität**	**PLZ**	**Erkrankung**
1	20≤Alter<30	*	13*	Herzerkrankung
2	20≤Alter<30	*	13*	Herzerkrankung
3	20≤Alter<30	*	13*	Virusinfektion
4	20≤Alter<30	*	13*	Virusinfektion
5	40≤Alter<60	*	14*	Krebs
6	40≤Alter<60	*	14*	Herzerkrankung
7	40≤Alter<60	*	14*	Virusinfektion
8	40≤Alter<60	*	14*	Virusinfektion
9	30≤Alter<40	*	13*	Krebs
10	30≤Alter<40	*	13*	Krebs
11	30≤Alter<40	*	13*	Krebs
12	30≤Alter<40	*	13*	Krebs

Tabelle 4.2: Generalisierte Patientendaten nach [MKGV07]

Im nun folgenden Schritt werden Datensätze mit ähnlichen nicht-sensiblen Attributen zusammengefasst um so Äquivalenzklassen bilden zu können, welche die Arbeit mit der Tabelle erleichtern.

	Nicht-sensible Attribute			Sensibles Attribut
Äquivalenz-klasse	**Alter**	**Nationalität**	**PLZ**	**Krankheit**
A	20≤Alter<30	*	13*	Herzerkrankung
				Herzerkrankung
				Virusinfektion
				Virusinfektion
B	40≤Alter<60	*	14*	Krebs
				Herzerkrankung
				Virusinfektion
				Virusinfektion
C	30≤Alter<40	*	13*	Krebs
				Krebs
				Krebs
				Krebs

Tabelle 4.3: 4-anonyme Beispieltabelle nach [MKGV07]

Leider bietet die *k-Anonymität* keinen vollständigen Datenschutz, wenn es darum geht, die Daten dauerhaft und vollständig anonym zu halten. Das Konzept der k-Anonymität weist bereits einige Mängel auf, die aber bereits in der Veröffentlichung von [Swe02] diskutiert wurden. Diese Mängel lassen es zu, das Daten deanonymisiert werden können, dies bedeutet, dass einzelne Elemente einer k-anonymen Tabelle unter bestimmten Umständen wieder eindeutig identifizierbar wären, was dem Grundgedanken des Datenschutzes widerspricht.

Zwei Ausnutzungsmöglichkeiten dieser Mängel werden im Folgenden kurz vorgestellt, um besser auf sie reagieren zu können bzw. sie von vornherein zu verhindern oder wenigstens zu erschweren.

Homogeneity Attack (nach [MKGV07])
Die sogenannten Homogenitäts-Attacke nutzt einen Umstand aus, unter dem alle k Datensätze einer Äquivalenzklasse identische sensible Attribute aufweisen. Weiß der Angreifer von der Existenz einer bestimmten Person in einer DB, und ist es ihm möglich, dieser Person eine Äquivalenzklasse zuzuweisen, so erfährt der Angreifer die sensiblen Attribute dieser Person.
Zum besseren Verständnis des Problems wird nun ein Beispiel zur Veranschaulichung gegeben.

	Nicht-sensible Attribute			Sensibles Attribut
Äquivalenz-klasse	**Alter**	**Nationalität**	**PLZ**	**Krankheit**
B	40≤Alter<60	*	14*	...
				Virusinfektion
C	30≤Alter<40	*	13*	Krebs
				Krebs
				Krebs
				Krebs

Tabelle 4.4: Ausschnitt aus Tabelle 4.3 für "Homogeneity Attack"

Alice und Bob sind Nachbarn und sich nicht gerade freundlich gesinnt, Alice ist zudem sehr neugierig. Eines Tages wird Bob krank und von der Ambulanz ins nächstgelegene Krankenhaus gefahren. Da Alice sehr neugierig ist, versucht sie herauszufinden, an was Bob erkrankt ist. Alice entdeckt die 4-anonyme Tabelle 4.4 mit aktuellen Patientendaten, die vom Krankenhaus veröffentlicht wurde. Sie weiß, dass Bobs Patientendaten in der Tabelle 4.4 enthalten sind und kennt aufgrund ihrer Nachbarschaft sein Alter (31), seine Nationalität (Amerikanisch) sowie seine Postleitzahl (13*). Dadurch schließt Alice darauf, das der Datensatz von Bob in der Äquivalenzklasse C enthalten sein muss und da alle Patienten dieser Äquivalenzklasse die gleiche Krankheit haben, erfährt Alice dadurch die von Bob.
Wie am Beispiel der in Tabelle 4.4 gegebenen Daten zu sehen ist, bietet die *k-Anonymität* keine vollständigen Datenschutz bei einer Homogenitäts-Attacke.

Als Lösung für diese Schwäche der *k-Anonymität* gegenüber der Homogenität-Attacke wurde von [MKGV07] ein anderes Verfahren, welches als *ℓ-Diversität* (oder auch *l-diversity*) bezeichnet wird, vorgestellt. Darauf aufbauend gibt den Lösungsansatz der sogenannten *t-closeness* [LLV07].

Background Knowledge Attack (nach [MKGV07] und [LY07])
Bei einem Angriff mittels Hintergrundwissen ist es möglich, Personen trotz k-Anonymität eindeutig zuzuordnen. Weiß der Angreifer von der Existenz einer Person in einer DB, ist es ihm möglich, auf Basis von Hintergrundwissen oder Zusatzwissen diese Person korrekt einer Äquivalenzklasse zuzuweisen, indem der Angreifer bestimmte sensible Attribute ausschließt oder besonders beachtet.
Auch hier wird zur Veranschaulichung und für das bessere Verständnis des Problems ein Beispiel gegeben.

	Nicht-sensible Attribute			Sensibles Attribut
Äquivalenz-klasse	**Alter**	**Nationalität**	**PLZ**	**Krankheit**
A	20≤Alter<30	*	13*	Herzerkrankung
				Herzerkrankung
				Virusinfektion
				Virusinfektion
B	40≤Alter<60	*	14*	Krebs
				...

Tabelle 4.5: Ausschnitt aus Tabelle 4.3 für "Background Knowledge Attack"

Alice hat eine Brieffreundin namens Umeko, die auch in ein Krankenhaus eingeliefert wird und deren Patientendaten in der 4-anonymen Tabelle 4.5 enthalten sind, welche vom Krankenhaus in regelmäßigen Abständen veröffentlicht wird. Alice weiß, das Umeko eine 21-jährige Japanerin ist, welche momentan unter der PLZ 13052 gemeldet ist. Basierend auf diesen Informationen über Umeko kann Alice nun schlussfolgern, das Umeko in der Äquivalenzklasse A der Patientendaten-Tabelle 4.5 enthalten sein muss.

Ohne zusätzliche Informationen kann Alice nun nicht sicher sagen, ob Umeko an einer Viruserkrankung oder einer Herzerkrankung leidet. Da jedoch hinlänglich bekannt ist, dass Japaner selten an Herzerkrankungen leiden, kann Alice über dieses Hintergrundwissen darauf schließen, das Umeko wohl an einer Viruserkrankung leidet.

Am Beispiel der Tabelle 4.5 ist zu sehen, dass *k-Anonymität* keinen vollständigen Datenschutz bei einer "Background Knowledge Attack" bietet.
Für diese Art des Angriffs wurde nach dem Stand dieser Arbeit bisher keine Lösung erarbeitet.

4.6 Präsentation der Empfehlung

Dieser Abschnitt des Konzeptes ist der Präsentation der Ergebnisses des Empfehlungssystems gewidmet. Dabei wird zuerst die Möglichkeit einer für den Nutzer erkennbare Einbindung von Empfehlungen vornehmen, dem wird die Möglichkeit der nahtlosen Integration gegenübergestellt.Abschließend für diesen Teil des Konzeptes werden einige Beispiele aus Portalen bzw. Websites vorgestellt, in denen Empfehlungen mit eingebunden wurden.

Die Präsentation des Ergebnisses des Recomendation Systems gehört mit zu schwierigsten Aspekten, wenn es darum geht, ein Empfehlungssystem mit einer Benutzeroberfläche (GUI) zu verknüpfen. Denn nicht selten ist die Nutzerakzeptanz gerade bei solchen Systemen schwierig einzuschätzen. Deshalb stellt sich die Frage: Muss die Einwirkung eines Empfehlungssystems überhaupt sichtbar gemacht werden? Oder reicht es nicht vollkommen aus, das Ergebnis des Empfehlungssystems nahtlos in die Präsentation der GUI mit einzubinden? Diese Frage soll nun diskutiert werden.

Wird eine Empfehlung in einer GUI präsentiert, so ergeben sich zwei Möglichkeiten der Darstellung. Es ist möglich, die Empfehlung als Empfehlung kenntlich zu machen oder aber die Empfehlung nahtlos mit einzubinden.
Geht man den Weg des Kenntlichmachens der Empfehlung, so erlaubt man dem Nutzer einen gewissen Einblick in das Empfehlungssystem und seine Ergebnisse. Dies kann positiv sein, da der Nutzer weiß, dass er mit seinen Daten für die Empfehlungsgebung und auch die Verbesserung der Selben wichtig ist.

Allerdings ist dies im aktuellen Kontext der Spionagearbeit von Geheimdiensten, der Aufklärung durch Edward Snowden und viele andere sowie durch das Thema des Datenschutzes allgemein ein sensibles Thema. Dies bedeutet, dass der Nutzer unbewusst oder bewusst in eine Reaktanz (vgl. [Ner08]) verfallen kann, da er dem Thema kritisch gegenüber steht und durch die Verwendung seiner Daten seine Persönlichkeitsrechte eingeschränkt sieht. Dem wäre entgegenzuwirken indem man den Nutzern im Portal zusichert, dass ihre sensiblen Daten geschützt und ausschließlich anonymisiert weiterverarbeitet werden, so wie es beispielsweise bei der Wegtam-Suche der Fall ist.
Verzichtet man gänzlich darauf Empfehlungen als solche zu kennzeichnen und lässt diese in das Gesamterlebnis mit einfließen, so kann man die Nutzerakzeptanz positiv beeinflussen, da es nicht offensichtlich ist, dass Nutzerdaten verwendet werden um Empfehlungen zu berechnen.
Der Mittelweg sollte sein, dass man Nutzer in den Nutzung-Vereinbarung des Portals darauf hinweist, dass Nutzer- und Nutzungsdaten genutzt werden, um Empfehlungen zu generieren und so das Erlebnis im Portal für ihn auch andere Nutzer positiv zu verbessern. Gleichzeitig sollte der Aspekt des Datenschutzes und der Anonymität hervorgehoben werden, um etwaige Abwehrreaktionen so gering wie möglich zu halten oder ganz zu vermeiden und dem Nutzer ein Gefühl der Sicherheit zu vermitteln.

Viele andere Portale bieten als einen Bestandteil des Inhaltes Empfehlungen an, dabei sollen im Folgenden einige wenige Beispiele präsentiert werden. Dies erscheint sinnvoll, da die Art und Weise der Präsentation von Empfehlungen mit Hilfe der aktuellen Möglichkeiten von “Liferay“ bzw. “Java“ sehr gut nachvollzogen werden kann. Gleichzeitig werden auch die Vor- und Nachteile der jeweiligen Präsentationen diskutiert.

Steam

Das Softwareportal Steam bietet eine recht einfache Variante einer Empfehlungspräsentation an.
Dabei wird auf der Hauptseite der Steam-Desktop-Clients ein kleiner Bereich der Startseite mit dem Label “Check out recommendations for you“ versehen, siehe Abbildung 4.6. Ebenso befindet sich ein Button “See recommendations“ direkt daneben, welcher eine Nutzerbetätigung erfordert.

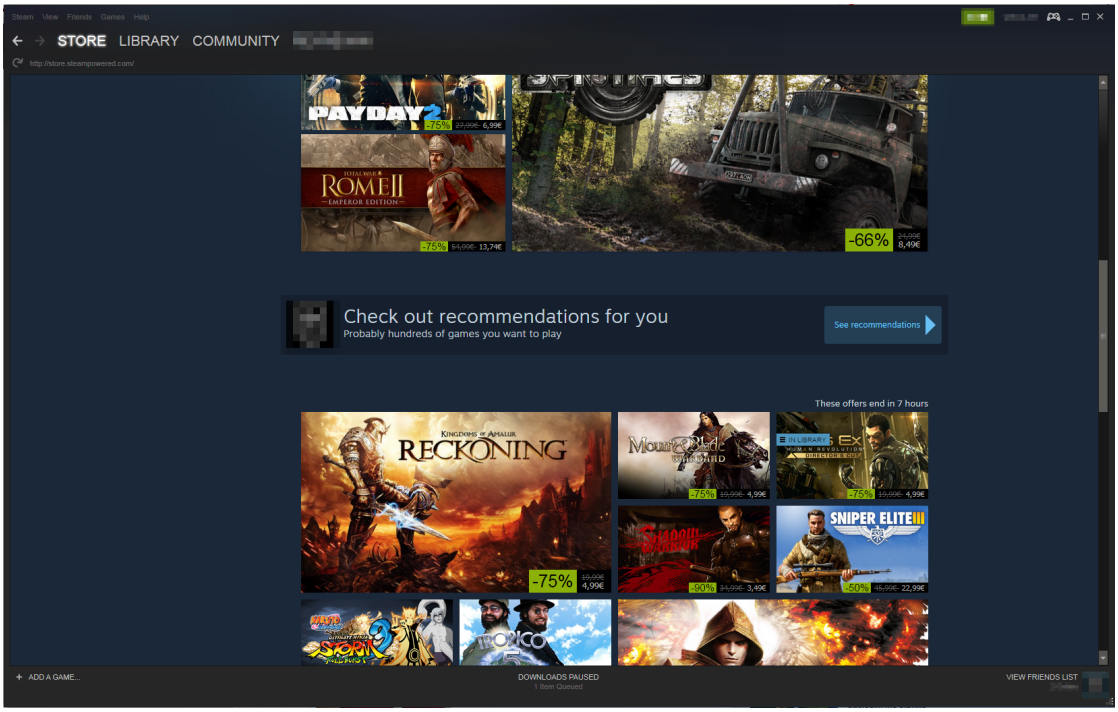

Abbildung 4.6: Steam-Empfehlungen auf der Startseite

Nach Betätigen des Buttons “See recommendations“ wird der Nutzer nun auf die eigentliche Empfehlungsseite weitergeleitet.

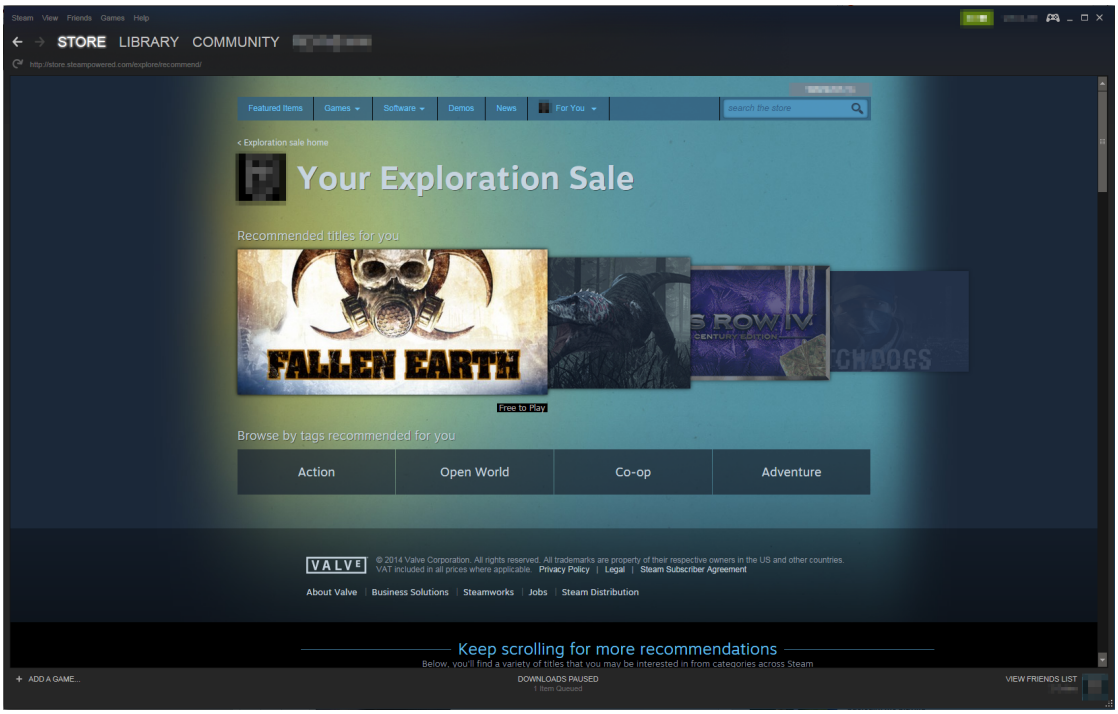

Abbildung 4.7: Steam Empfehlungsseite

Auf dieser Seite sind nun einige Spiele als Empfehlungen zu sehen. Dabei werden die Spielen in einem nach links verlaufenden Stapel versetzt zueinander nach rechts angezeigt, wobei nur mit dem vordersten Spiel ganz links interagiert werden kann. Zur Interaktion mit den anderen Empfehlungen müssen diese über Mauseingaben angesteuert (gescrollt) werden. Zusätzlich ist in Abbildung 4.7 unterhalb der Empfehlungen eine weitere Möglichkeit zur besseren und genaueren Empfehlungsberechnung zu sehen. Dabei handelt es sich um sogenannte “Tags“ mit deren Hilfe ein genaueres Nutzerprofil erstellt werden kann, um so bessere Empfehlungen geben zu können.
Am unteren Ende von Abbildung 4.7 ist eine weitere Möglichkeit von Empfehlungen zu sehen. Dort werden Empfehlungen nach Gruppen, in diesem Falle Arten von Spielen, sortiert und ausgegeben. Diese Empfehlungen werden erst auf Anforderung geladen, sobald man mit der Maus herunterscrollt.

Insgesamt fällt bei dem Empfehlungssystem von Steam auf, dass es Nutzereingaben erfordert, um zur eigentlichen Empfehlungsseite zu gelangen, und um für weitere Empfehlungen ein genaueres Profil zu erstellen. Hier ist die Nutzerinteraktion notwendig, um zu guten Empfehlungen zu gelangen. Dies kann vom Nutzer positiv aber auch negativ bewertet werden. Positiv an dieser Variante ist die Offenheit der Präsentation der Empfehlung, da Empfehlungen nicht einfach in den Client integriert werden, sondern dass der Nutzer ein aktives Mitspracherecht durch Ignorieren oder Wahrnehmen dieser Empfehlungen hat. So wirken die Empfehlungen nie aufgedrängt, da man sich als Nutzer selber zum Aufrufen des Empfehlungssystems entschieden hat. Damit umgeht Steam geschickt Abwehrreaktionen, da Nutzer keine Empfehlungen wahrnehmen müssen.
Negativ an dieser Art der Präsentation ist natürlich der erhöhte Aufwand den der Nutzer betreiben muss, sofern er denn Empfehlungen erhalten will, dennoch ist dieser Mehraufwand vertretbar und ist relativ zu den positiven Aspekten vergleichsweise gering.

Facebook

Ein weiteres populäres Beispiel für ein aktuelles Portal ist *Facebook*. Es wird als sogenanntes "soziales Netzwerk" beschrieben, welches eine Form der Netzgemeinschaften darstellt, die technisch durch Webanwendungen sowie Portale abgebildet werden können. Dadurch kann man ein soziales Netzwerk als eine Unterart von Portalen definieren. Facebook nutzt beispielsweise zur Personalisierung der Inhalte die Profile von Nutzern, um diesen gezielt Vorschläge (Empfehlungen) für diverse Themen, Gruppen, etc., geben zu können. Hierbei soll nun genauer auf die Präsentation von Gruppenempfehlungen eingegangen werden.

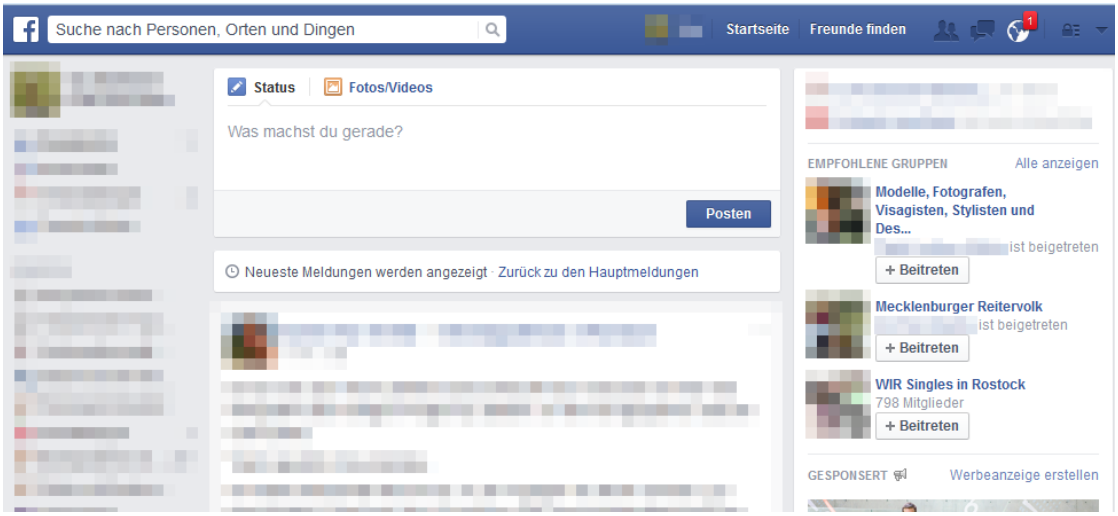

Abbildung 4.8: Facebook Gruppenempfehlung, normale Nutzung

Facebook bietet, wie auf Abbildung 4.8 zu sehen ist, Gruppenvorschläge im normalen Nutzungsfluss in einer Spalte, für gewöhnlich rechts des Hauptinhaltes der jeweiligen Seite an. An dieser Stelle kann auch Werbung stehen, für das Portal *myKOSMOS* hingegen ist dies recht unsinnig. Die Länge der Gruppenempfehlung variiert normalerweise, beträgt aber immer mindestens 2 Gruppen, für gewöhnlich sind es um die 5 Empfehlungen. Klickt man nun auf das Feld “Alle anzeigen“, so wird man auf eine Seite weitergeleitet, die sich ausschließlich der Gruppenempfehlung widmet, siehe Abbildung 4.9.

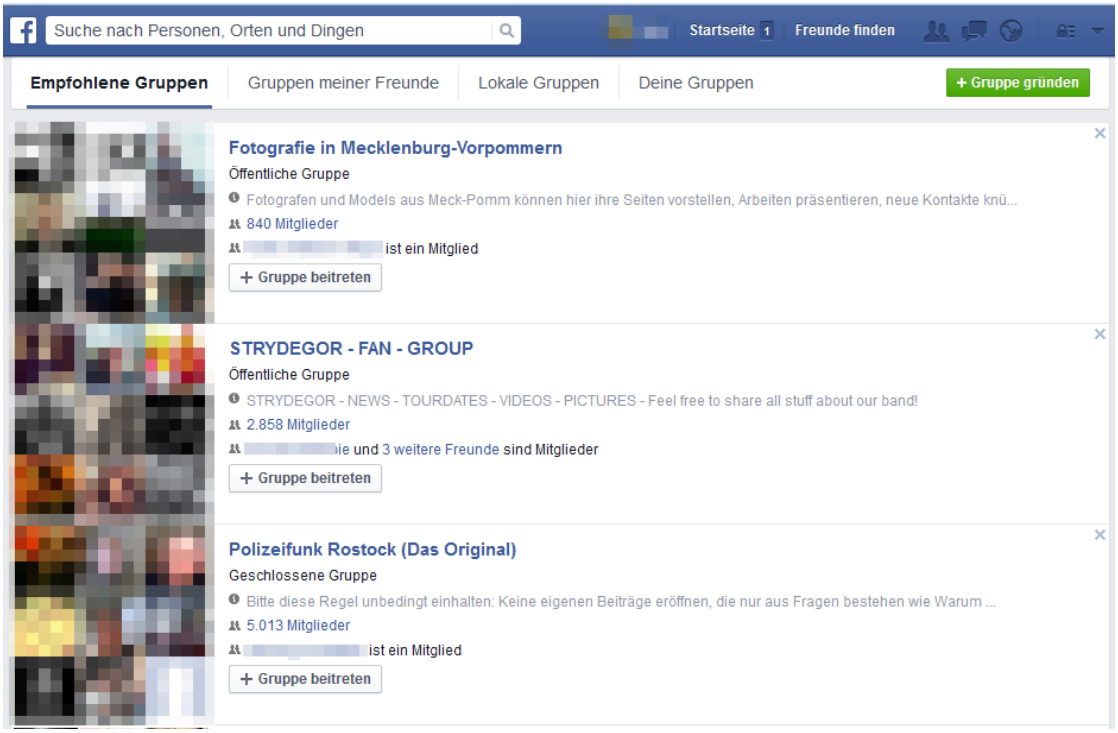

Abbildung 4.9: Facebook Gruppenempfehlung, Empfehlungsseite

Auf der eigentlichen Empfehlungsseite ergibt sich eine bessere Übersicht über die Gruppenempfehlungen. Dabei wird, im Gegensatz zur kleineren Anzeige am Rand, deutlicher, dass die Empfehlungen auf Freunden basieren, sowie auch auf der Mitgliedschaft in anderen Gruppen. Generell gesprochen werden die Empfehlungen nach dem Ähnlichkeitsprinzip ausgegeben.

Positiv an der gesamten Präsentation ist, dass sie den Nutzer während der Nutzung im Portal selber nicht stört. Sie wird wie bei Steam auch als kleine Auswahl neben dem eigentlichen Hauptteil im Portal angeboten. Der Nutzer kann diese Empfehlungen auch bewusst ignorieren, da sie wie schon beschrieben nahtlos integriert sind (siehe Abbildung 4.8). Wünscht der Nutzer dennoch eine größere Auswahl an Empfehlungen, so kann er diese gezielt aufrufen (siehe Abbildung 4.9).

Erst hier werden die Empfehlungsdaten detaillierter und der Nutzer erfährt (teilweise), wie die Empfehlungsgebung erfolgt. Auch hier wird der Nutzer langsam und durch seinen Willen an die Empfehlungen herangeführt, was ebenfalls Abwehrreaktionen massiv reduzieren kann. Diese Art der Empfehlungspräsentation würde sehr gut in das Portalkonzept von “myKOSMOS“ passen.
Negativ ist hieran lediglich der “Aufwand“ um zur detaillierten Empfehlungsseite zu gelangen.

Google

Wie bereits in Abschnitt 4.2 in Abbildung 4.4 gezeigt, besitzt Google bereits bei der Eingabe eines Suchbegriffes die Möglichkeit, ein Empfehlungssystems zum Einsatz zubringen, in dem eventuell zu dem Suchbegriff passende Ergänzungen vorgeschlagen werden. Dies wäre hervorragend für die Suchfunktion des Portals nutzbar, um den Nutzer aktiv bei der Suche zu unterstützen. Auch wäre es möglich, dem Nutzer je nach Begriff eine Sammlung von Seiten zur Durchsuchung zu empfehlen, wie es “Wegtam“ bereits unterstützt.
Positiv daran ist direkte und aktive Unterstützung des Nutzer bei der Sucheingabe, da diese bei Bedarf genutzt oder ignoriert werden kann. Nachteilig wäre der hohe Aufwand für Berechnung bei Suchbegriffen. Hier sollte aber wie bereits erwähnt, eine Liste von passenden Begriffen als Empfehlung im Vorfeld berechnet werden, die in periodischen Abständen aktualisiert wird.

Ebenso besitzt Google die Möglichkeit, die Suchergebnisse zu sortieren, um so das Ergebnis für den Nutzer zu personalisieren. Auch dies wäre von der Portalsuche hervorragend nutzbar, sowohl auf der Ebene der Suchergebnisse als auch auf der Ebene der Filter für die Suchergebnisse. Hier könnten solche Filter empfohlen werden. Für die Empfehlungsgebung bei den Ergebnissen direkt müssten diese abgefangen und entsprechend neu sortiert werden, daher wäre es sinnvoller für den Beginn der Nutzung des Portals mit Filterempfehlungen zu arbeiten.
Positiv ist hierbei, dass der Nutzer über die Sortierung im Unklaren gelassen wird, was bei normaler Nutzung vom Nutzer nicht wahrgenommen wird. Dies senkt von vornherein die Wahrscheinlichkeit von Abwehrreaktionen, denn wenn der Nutzer nicht über eine Empfehlung unterrichtet wird, wie wäre es ihm dann möglich darauf zu reagieren.

Negativ hingegen könnten eventuelle Abwehrreaktionen des Nutzers sein, sollte er diese Umsortierung realisieren. Dem kann aber, wie bereits erläutert, durch eine entsprechende Formulierung in der Nutzungsvereinbarung des Portals entgegengesteuert werden, indem man den Nutzer von vornherein über das Empfehlungssystem und die Nutzung seiner Daten dafür aufklärt und ihm gleichzeitig Datenschutz zusichert.

4.7 Bewertung von gegebenen Empfehlungen

In diesem Abschnitt des Konzeptes wird auf die Notwendigkeit eingegangen, dass gegebene Empfehlungen auf ihre Qualität hin überprüft werden können und auch müssen, um einen gewissen Standard zu erfüllen. Dies betrifft im Besonderen die Präzision der Empfehlung, aber auch die allgemeine Präsentation der Selbigen. Die Bewertung von gegebene Empfehlungen nimmt einen eigenen Abschnitt ein, da sie nicht thematisch passend in einem anderen Abschnitt des Konzeptes verortet werden konnte. Zudem wäre es der Wichtigkeit des Themas und der Problematik nicht gerecht geworden, die Bewertung von Empfehlungen in einen anderen Abschnitt mit einzubinden.

Grundsätzlich lässt sich sagen, dass es ohne eine Überprüfung der Qualität der Empfehlungen schwierig ist, die Nutzerakzeptanz festzustellen und dort gegebenenfalls entsprechend steuernd einzugreifen. Ebenso muss es einen Weg geben, um die Genauigkeit der gegebenen Empfehlungen zu überprüfen, um auch dort eingreifen zu können. Deshalb ist es wichtig eine oder mehrere Möglichkeiten zu finden, um die Qualität der Empfehlungen zu überprüfen, um so einen Standard sichern zu können. Wie dies geschehen kann, wird im folgenden diskutiert.

Eine Möglichkeit zur Überprüfung der Qualität wäre die direkte Befragung der Nutzer, sei es durch eine portalweite Umfrage oder aber besser direkt bei jeder Empfehlung, siehe Abbildung 4.10.

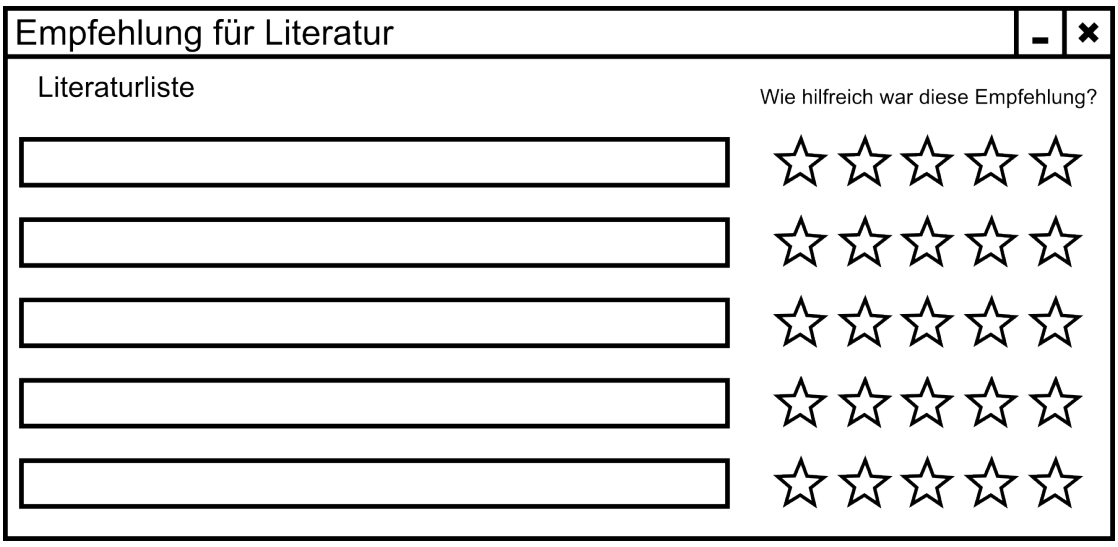

Abbildung 4.10: Bewertung der Empfehlung durch den Nutzer (eigene Darstellung)

Dies hätte den Vorteil, dass sich der Nutzer aktiv und bewusst mit seiner Bewertung zusätzlich in das Empfehlungssystem und das Portal einbringen kann und so auch wahrnimmt, das er bewusst aktiv steuernd eingreifen kann. Natürlich kann dies auch von Nachteil sein, dass der Nutzer nun gezielt darüber informiert wird, dass er (also sein Nutzerprofil), für die Empfehlungsberechnung eingesetzt wird. Ein weiterer Nachteil, der sich aus dem vorherigen ergibt, ist, dass der Nutzer in eine sogenannte Reaktanz (vgl. [Ner08]) verfallen kann, das bedeutet, das der Nutzer das Empfehlungssystem aus welchen Gründen auch immer, nicht akzeptieren will, und sich deshalb bewusst oder unbewusst gegen eine Bewertung sperrt. Dies kann natürlich auch ohne eine Reaktanz erfolgen, wenn der Nutzer einfach keine Notwendigkeit sieht oder keine Motivation verspürt, eine Bewertung für die Empfehlung abzugeben. Deswegen wäre es im Sinne der Sicherung des Qualitätsstandards der Empfehlungen sinnvoll, den Nutzer zu motivieren, eine Bewertung abzugeben und die Reaktanz so gering wie möglich zu halten.

Eine weitere Möglichkeit ist, den Nutzer über die Existenz und die Auswirkung des Empfehlungssystems weitestgehend im unklaren zu lassen. Dies bedeutet das zwar Empfehlungen generiert werden, diese aber nicht als solche kenntlich gemacht werden.

Bei dieser Methode ist der Aufwand der Qualitätsmessung relativ zur ersten Variante mit direkter aktiver Nutzerbeteiligung größer, aber die Möglichkeit einer Reaktanz wird massiv verringert. Um nun ohne direkter Nutzerbeteiligung zu Ergebnissen für den Qualitätsstandard zu kommen, gibt es verschiedene Wege.

Eine Möglichkeit wäre, Empfehlungen komplett, d.h. mit *usern* und empfohlenen *items*, abzuspeichern und dann beispielsweise über die Aktivierung eines Links zu messen, ob diese oder andere Empfehlung genutzt wurden. Daraus ließe sich, entsprechende Nutzerbeteiligung vorausgesetzt, sehr schnell ein Bild gewinnen, welche **items** "empfehlungswürdiger" als andere sind. Ebenso lassen sich daraus unpassende **items** herausfiltern, wobei hier auch wiederum die Schwierigkeit besteht, genau zu messen, ob das **item** wirklich als Empfehlung "nutzlos" ist oder aber einfach nur nicht wahrgenommen wird.
Eine Möglichkeit, dies festzustellen ist, dass man für erste Phase der Überprüfung einer Empfehlung *item*-weise Vergleiche anstellt, um einen Überblick über deren Popularität, d.h. deren Genauigkeit, zu bekommen. Hierbei ist es sinnvoll **items** mit einer Prioritätsliste zu versehen, um so dem Empfehlungssystem eine Hilfestellung zu bieten.
Die zweite Phase sollte nun, basierend auf den Messergebnissen, eine Umsortierung von "weniger populär" nach "sehr populär" stattfinden, um so die Rangfolge zugunsten der weniger populären *items* zu verändern. Sollte sich nach dieser Phase herausstellen, dass die Popularität einiger *items* signifikant hoch oder niedrig bleibt, so sollten diese *items* entsprechend behandelt werden und eine Umwertung der Priorität stattfinden. Ebenso sollte mit den anderen *items* verfahren werden, deren Popularität sich verändert hat.
Nach dem Ende der zweiten Phase hat sich mit hoher Wahrscheinlichkeit herauskristallisiert, welche *items* nun eine hoher Nutzungsrate aufweisen, also eine gute Empfehlung waren, und auf welche *items* dies nicht zutrifft. Natürlich muss diese Messung in regelmäßigen Abständen, wenn nicht gar nach jeder Nutzerinteraktion ausgeführt werden, um ein möglichst aktuelles Bild von "empfehlungswürdigen" *items* zu erhalten. Zudem muss, um Verzerrungen zu vermeiden, bei *items*, die neu in den Datenbestand aufgenommen werden, diese Prozedur von vorn durchlaufen werden, oder aber den neuen *items* wird zu Beginn die höchstmögliche Priorität eingeräumt, um schnell herauszufinden, ob es sich um ein weiteres empfehlungswürdiges *item* handelt. Denkbar und sehr sinnvoll wäre eine Verbindung der genannten Methoden, da dies den Aufwand geringfügig, aber die Empfehlungsgüte stärker erhöhen würde.

Eine andere Möglichkeit der Auswertung wäre die Überwachung von Interaktionen und Bewegungen des Nutzers im Portal. Dies könnte durch die Überwachung der Mausbewegung geschehen, im Sinne des sogenannten Canvas Fingerprinting.
Canvas Fingerprinting ist eine Form von sogenannten "User-Tracking"-Techniken, das wohl populärste Beispiel für eine solche Technik sind die "Cookies'", welche üblicherweise Daten von und über besuchte Websites speichern. Dies kann positive Ausprägungen haben, z.B. kann sich so ein Webbrowser ohne Aufwand in eine bereits besuchte verschlüsselte Seite einloggen. Eine negative Form der Anwendung erhalten Cookies, wenn sie dafür genutzt werden, differenzierte Nutzerdaten über das Surfverhalten an Dritte senden, ohne das der Nutzer dies will.
Canvas Fingerprinting verhält sich anders als es Cookies tun, da für das Canvas Fingerprinting keine zusätzlichen Daten auf Seite des Nutzers erstellt werden. Stattdessen wird die HTML5-Darstellung der sogenannten Canvas-Elemente ausgenutzt, da diese je nach Konfiguration des Computers (GPU, GPU-Treiber, Browser, Schriftart) anders (man kann von einzigartig sprechen) dargestellt werden. Dazu wird meist ein verborgenes Bild gezeichnet, welches es ermöglicht einem bestimmten Browser eine ID zuzuweisen, mit der er eindeutig wiedererkannt werden kann [can15]. Ein Nachteil dieser Technik ergibt sich, sobald beispielsweise die Schriftart geändert wird, nun kann der Computer nicht mehr eindeutig zugeordnet werden. Dennoch wurde im Rahmen der Studie von [Eck10] festgestellt, dass eine eindeutige Zuordnung von 83,6% gegeben ist.
Diese Technik könnte, in Abwandlung zu positiven Zwecken, für das Empfehlungssystem eingesetzt werden, um Daten über die Qualität der gegebenen Empfehlungen zu erhalten. Dies würde den Auswertungsaufwand massiv erhöhen, es sei denn man würde ein selbstlernendes System verwenden, was auch gleichzeitig die Berechnung und Auswertung übernimmt, und diese Ergebnisse dann an das eigentlich Empfehlungssystem weitergibt. Allerdings stellt auch dieses selbstlernende System einen hohen Aufwand dar, weswegen der Aufwand der Implementierung eines solchen Systems zu seinem Nutzen im zeitlichen Kontext evaluiert werden muss.

4.8 Bewertung des Konzeptes

Abschließend für dieses Kapitel wird in diesem Abschnitt die Bewertung des erstellten Konzeptes zur Anbindung eines Empfehlungssystems an das Portal *myKOSMOS* enthalten sein.

Bei der Konzeption des Empfehlungssystems wurde zuerst von der Definition des Einsatzzweckes ausgegangen. Dadurch ließen sich bereits vergleichbar große Einsatzfelder für ein solches Empfehlungssystem finden. Dies ist insofern als positiv zu bewerten, als dass durch diese Felder bereits ein erster Eindruck über die Größe, die Priorität und den Umfang der Einsatzbereiche gefunden werden konnte. Ebenso können so durch Umkehrung Einsatzfelder ausgeschlossen oder in der Priorität entsprechend versetzt werden. Auf der anderen Seite sind die so abgesteckten Bereiche recht groß und auch stellenweise nicht scharf definiert. Deswegen war es nötig, diese Bereiche genauer zu untersuchen um gezielte Einsatzpunkte in diesen Bereichen zu finden. Diese wurden zu einem Teil am Ende des Abschnitts 4.1 dargestellt, der größere Teil wurde in Abschnitt 4.2 genauer untersucht. Dabei stellte sich heraus, dass je tiefer und genauer man diese Bereiche untersucht, diese immer umfangreicher und granular feiner werden. Deswegen wurde entschieden, um dem Umfang und dem Zeitfaktor einer Bachelor-Arbeit gerecht zu bleiben, das ab einer bestimmten Ebene nicht feiner strukturiert wird und dafür ein kurzer Blick über das gerade verfügbare Feld genügen muss.

Im Rahmen der Konzeption aus Sicht der Prozessorientierung konnten die im ersten Abschnitt 4.1 gefundenen Einsatzgebiete entsprechend durch Prozessmodellierung feiner unterteilt werden, um so mögliche empfehlungswürdige **items** zu finden. Dazu wurden mehrere Modelle von Studenten erstellt, welche ein Semester an der Universität Rostock mit Nutzung des Portals absolvieren. Diese Modelle wurden als Skizzen konzipiert, um einen groben Überblick über den Ablauf eines Semesters zu geben. Dabei wurden unnötige bzw. selbsterklärende Prozessbestandteile bewusst nicht integriert, um die Modelle übersichtlich zu halten. Dies ist gut gelungen, dennoch gingen im Rahmen dieser Verallgemeinerungen auch kleine Schritte verloren, welche dann bei den identifizierten **items** wieder aufgegriffen wurden.

Auch hier stellte sich heraus, dass es, je genauer man die Prozessmodellierung betreibt, mehr und mehr unterschiedliche Strukturen (**items**) gibt, die empfehlungswürdig sind, weswegen, um eine Übersichtlichkeit zu gewährleisten, nicht mehr genauer untersucht und unterteilt wurde, da auch hier wieder der Umfang und der Faktor Zeit eine entscheidene Rolle spielten. Die Liste der herausgearbeiteten **items** erhebt keine Anspruch auf Vollständigkeit und sollte im Rahmen weiterer Arbeiten erweitert und aktualisiert werden. Dennoch ließen sich viele der in Abschnitt 4.2 genannten **items** mit mehr oder weniger Aufwand herausarbeiten und entsprechend beschreiben. Die empfehlungswürdigen **items** wurden beschrieben, ebenso wie die Empfehlungsgebung mit und zu diesen ablaufen soll. Abschließend für den Abschnitt 4.2 wurden die herausgearbeiteten **items** am Ende des Abschnittes noch einmal kurz zusammengefasst und beschrieben, um dem Leser noch einmal die Möglichkeit zu geben, einen schnellen Überblick über behandelte **items** zu geben.

In Abschnitt 4.3 wurde die Konzeption im Rahmen der technischen Referenz von “Liferay“ fortgeführt. Hierbei wurde zuerst aus Sicht der Grundinstallation von “Liferay“ gearbeitet, um einen Überblick über bereits für ein Empfehlungssystem nutzbare Datenstrukturen zu bekommen. Wie bereits in den oberen Absätzen beschrieben, führte auch hier die Einarbeitung in die Tiefe zu einem mehr als umfangreichen Endergebnis, weswegen auch hier ab einer bestimmten Tiefe beschlossen wurde, abzubrechen und nur einen kurzen Überblick zu geben. Dies hat mehrere Gründe, zum einen ist die Dokumentation zu “Liferay“ an vielen notwendigen Stellen gar nicht oder nur rudimentär vorhanden (diese Aussage bezieht sich auf das “Liferay-Wiki“). Auch sind Strukturen nicht immer eingängig gestaltet, sodass es relativ viel Aufwand erfordert sich in selbige einzuarbeiten. Das bedeutet, dass viel Aufwand in das Suchen und dann das Erstellen der Dokumentation für die eigene Nutzung gesteckt werden muss. Auf der anderen Seite bietet “Liferay“ die Möglichkeit, sich den Quelltext (Source Code) und die Dokumentation des Selbigen (JavaDocs), zumindest für die “Community Edition“, herunter zu laden. Diese Pakete umfassen zusammen, allerdings noch gepackt, schon eine Größe von etwa 450 MB (LifeRay CE 6.2.0), entpackt enthalten alleine die JavaDocs bereits knappe 50.000 Dateien mit einer Gesamtgröße von etwa 2,10 GB, siehe Abbildung 4.11.

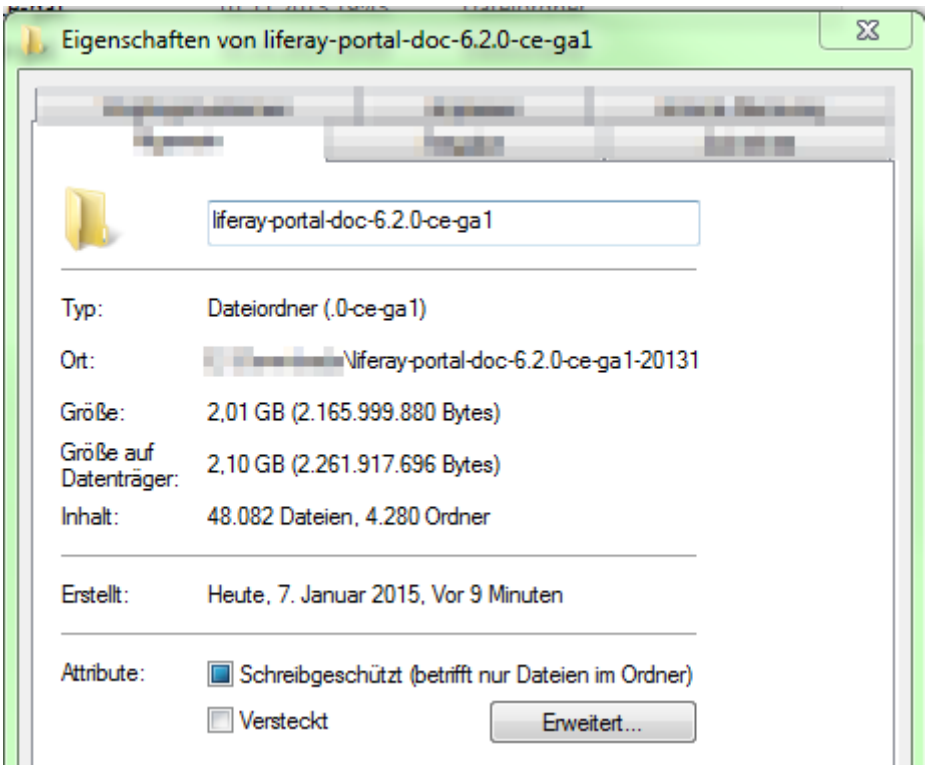

Abbildung 4.11: Liferay JavaDocs entpackt

Unter diesem Aspekt wurde auf eine genauere Einarbeitung verzichtet, weswegen die Beschreibung nur die wichtigsten Komponenten und deren Datenstrukturen nach einer Liferay-Neuinstallation enthält. Deswegen ist dieser Teil nicht vollständig. Dies ist aber auch gar nicht notwendig, da sich herausstellte, dass Liferay selber nur relativ wenige, gemessen am Gesamtumfang, für ein Empfehlungssystem nutzbare Datenstrukturen mitbringt. Eine weitere Einarbeitung wäre aber dennoch sinnvoll, sofern es zeitlich möglich ist, um das Portal, Liferay und die Zusammenhänge zwischen den einzelnen Portlets besser zu verstehen und dann entsprechend anwenden zu können.
Im folgenden Teil des Abschnittes wurde auf die bereits für das Portal “myKOSMOS“ entwickelten Portlets und deren Datenstrukturen genauer eingegangen. Dabei stellte sich heraus, dass das “Support2-Portlet“, das “Skype Chat & IM-Portlet“ sowie das in der Entwicklung befindliche “StudIP-Portlet“ viele Verbesserungen mitbringen, von denen jene des “Support2-Portlets“ perfekt für das zukünftige Empfehlungssystem nutzbar sind.

Dabei geht es einmal um die Beobachterkomponente, aber vorrangig um die Datenstruktur, die jene Komponente nutzt und befüllt. Diese Struktur sollte vom zukünftigen Empfehlungssystem genutzt werden, dazu ist es allerdings auch notwendig, diese Datenstruktur zu erweitern und zu aktualisieren.
Daran anschließend wurde kurz auf die Bedeutung der "Liferay Hooks" eingegangen. Diese können von einer Beobachtungskomponente dazu genutzt werden, um Daten über Aktivitäten im Portal zu sammeln. Das Empfehlungssystem kann anschließend auf diese Daten zugreifen und diese entsprechend verarbeiten. Dies ist ein wichtiger Punkt, der in der zukünftigen Konzeption einer Bobachtungskomponente berücksichtigt werden sollte.
Bevor auch in diesem Abschnitt des Konzeptes eine Zusammenfassung geben wurde, wurde zunächst auf die in Abschnitt 4.2 bereits herausgearbeiteten **items** eingegangen. Dazu wurden notwendige Datenstrukturen herausgearbeitet, die für die Verwendung der **items** in einem Empfehlungssystem notwendig sind. Dies geschah nutzungsorientiert, um nur die für eine Verarbeitung durch ein Empfehlungssystem notwendigen *Attribute* mit einzuarbeiten. Natürlich müssen diese für eine spätere Nutzung und Präsentation entsprechend integriert werden.

Der Abschnitt 4.4 enthält wünschenswerte Erweiterungen des Portals, welche dem Empfehlungssystem zukünftig helfen sollen, die Empfehlungsgebung von vornherein zu verbessern. Die Gruppenfunktion beispielsweise würde einen Teil der notwendigen Aggregation zu Empfehlungsgruppen vereinfachen, da hier schon der Nutzer eine Teil der Arbeit selbst übernommen hat. Die dafür notwendigen Datenstrukturen sind vorhanden, allerdings muss die Gruppenerzeugung auch auf Nutzerebene möglich sein. Bisher ist dies nur auf Administrator-Ebene möglich. Cloud-Dienste stellen ebenfalls eine Erweiterung dar, die von Empfehlungssystemen genutzt werden kann. Dies hat den Hintergrund, dass in einem Cloud-Dienst Dateien zentral gelagert und verbreitet werden können. Dies stellt natürlich einen gewissen Aufwand dar, dennoch würde das Portal insgesamt und das Empfehlungssystem an sich davon profitieren. Ebenso verhält es sich mit der Suche. Diese sollte um Möglichkeiten der Beobachtung und der Empfehlungsgebung erweitert werden, um dem Nutzer im Portal zu helfen. Ähnlich gelagert ist der Fall der vorgeschlagenen Erweiterung des Portals um gastronomische Einrichtungen. Dabei sollte diesem Punkt eine geringere Priorität bei der Umsetzung eingeräumt werden, da diese Erweiterung weniger der akademischen Nutzung dient, sondern eher der Befriedigung persönlicher Bedürfnisse.

Da Liferay auch über die Möglichkeit der Integration eines Portal-Shops verfügt, wurde diese Erweiterungsoption ebenfalls mit aufgenommen. Vielleicht ist es möglich, diese Erweiterung als kleines Arbeitspaket im Rahmen der Arbeit am Portal umzusetzen.

Bestandteil des Konzeptes musste ebenfalls der Datenschutzes (Abschnitt 4.5) sein, da ein Empfehlungssystem immer zu einem gewissen Teil mit sensiblen Daten umgehen muss, um Empfehlungen zu berechnen. Dabei wurde der sogenannten "k-Anonymität" besondere Bedeutung zugemessen, da ein Empfehlungssystem besonders mit aggregierten Daten arbeiten muss, um nicht nur Nutzer, sondern auch Gruppen von Nutzern entsprechend mit Empfehlungen bedienen zu können. Die "k-Anonymität" sichert jeder Person in einer aggregierten Gruppe ein gewisses Maß an Anonymität zu, allerdings ist auch dieses Konzept nicht ohne Fehler, weswegen zwei mögliche Varianten beschrieben wurden, bei denen trotz k-Anonymität sensible Attribute Personen zugeordnet werden konnten. Dies diente dazu, Entwicklern in der Zukunft diese Probleme bewusst zu machen, so dass Lösungsstrategien dafür entwickelt werden können. Für die "Homogeneity Attack" wurden dabei zwei Lösungsmöglichkeiten aufgezeigt.

Im Abschnitt 4.6 wurde auf die Notwendigkeit einer guten grafischen Präsentation einer Empfehlung eingegangen. Dazu wurde zuerst die Möglichkeiten der Reaktion von Nutzern auf die "offene" und die "nicht vorhandene" Präsentation einer Empfehlung diskutiert. Jede Seite hatte ihre Vor- und Nachteile. Am Ende ergab sich die Lösung, Empfehlungen vorrangig "verdeckt" zu platzieren, sie also nahtlos in das Portalerlebnis zu integrieren. Dies erschwert natürlich die in Abschnitt 4.7 erörterte Notwendigkeit, Empfehlungen auf ihre Qualität hin zu überprüfen. Dennoch wären auf die Weise Abwehrreaktionen von Nutzern zu einem großen Teil vermeidbar.
Abschließend für diesen Abschnitt des Konzeptes wurden andere Portale und Websites auf die Art und Weise untersucht, wie dort Empfehlungsgebung geschehen. Dies war notwendig um eine kleine Übersicht über verschiedene Präsentationsmöglichkeiten einer Empfehlungen zu geben. Dabei zeigte sich, dass je nach Zielgruppe, offene wie auch verdeckte Empfehlungen ihren Zweck erfüllen können und dennoch eine große Nutzerakzeptanz erfahren.

Abschließend für das Konzept wurde in Abschnitt 4.8 die Möglichkeit der Bewertung von Empfehlungen diskutiert. Die Möglichkeit festzustellen, wie gut oder schlecht eine Empfehlung ist oder war, ist für die Einhaltung eines gewissen Qualitätsstandards bei der Empfehlungsgebung von entscheidender Bedeutung. Wie sonst sollte es möglich sein, die Qualität von Empfehlungen direkt und genau zurechenbar zu überprüfen? Natürlich bedeutet dies eine erhöhten Aufwand bei der Umsetzung eines Empfehlungssystems, allerdings können so gleich von Beginn der Nutzung des Portals an, Daten über die Genauigkeit der Empfehlungen getroffen werden und so die Berechnungsparameter entsprechend angepasst werden, um schneller zu besseren Empfehlungen zu gelangen.

All dies trägt zu einer größeren Nutzerakzeptanz des Portals und des Empfehlungssystems bei.

5 Implementierung des Empfehlungssystems

Dieses Kapitel geht auf die prototypische Implementierung eines Teils des Konzeptes ein. Hierfür wurde als Teil des Konzeptes die Wahl von Wahlpflichtfächern ausgewählt. Im ersten Abschnitt dieses Kapitels wird die für diese Implementierung verwendete Software kurz erläutert. Im zweiten Abschnitt wird der Ablauf des Programmes an sich dargestellt und in den einzelnen Phasen erläutert, was in diesen im Programm geschieht. Abschließend für dieses Kapitel wird eine Bewertung der Implementierung vorgenommen, in der die für diese Arbeit notwendigen Änderungen erläutert werden.

5.1 Verwendete Software

Dieser Abschnitt beschreibt die für die Implementierung notwendige und verwendete Software. Dabei wird nur eine kurze Übersicht gegeben, da nicht die genaue Funktionsweise der verwendeten Software ausschlaggebend ist.

Empfehlungssystem von [Ihl14]

Für die prototypische Implementierung eines Teils des Konzeptes aus Kapitel 4 wird das bereits entwickelte Empfehlungssystem aus der Bachelor-Arbeit von [Ihl14] genutzt. Dieses Empfehlungssystem ist zwar noch keine vollständige Software, dennoch verfügt sie bereits über wichtige Komponenten wie den hybriden Ansatz für die Empfehlungsberechnung. Für die Zwecke dieser Arbeit werden Teile der Software entsprechend umgeschrieben, so dass sie den Ansprüchen für die Implementierung in dieser Arbeit genügen.

Liferay

Wie bereits in Kapitel 2 in Abschnitt 2.3 erklärt, nutzt *myKOSMOS* zur Dar- und Erstellung des Portals die Portal-Software *Liferay*. Diese ermöglicht es, mittels *JavaServerPages* (.jsp) sogenannte Portlets zu erstellen, um Funktionalitäten zu bündeln. Ziel ist es deshalb ein Empfehlungssystem in diese Portlets zu integrieren, um dem Nutzer des Portals auf ihn zugeschnittene Literatur (generell Lernmaterialien), Lerngruppen, Informationen über Vorlesungen (generell Lehrveranstaltungen) oder andere empfehlungswürdige **items** anzubieten.

Eclipse

Eclipse ist ein quelloffenes Programmiertool welches zur Entwicklung von Software unterschiedlichster Art eingesetzt wird. Gegründet wurde das "Eclipse-Projekt" 2001 von IBM und war als integrierte Entwicklungsumgebung (IDE) für die Programmiersprache Java gedacht [ecl14a]. Da *Eclipse* sehr gut erweiterbar ist, wird es mittlerweile auch für andere Entwicklungsaufgaben eingesetzt, beispielsweise als Entwicklungsumgebung für C++ oder PHP Projekte [ecl14b]. Für *Eclipse* gibt es ebenfalls eine Vielzahl sowohl quelloffener wie auch kommerzieller Erweiterungen, wie z.B. *PyDev* (open source) für die Programmiersprache Python [pyd14].

MySQL

MySQL ist ein quelloffenes SQL client/server Datenbankmanagementsystem (DBMS), welches ursprünglich aus Skandinavien stammt. Die Wurzeln dieser Software liegen im Datenbankwerkzeug UNIREG, welches von Michael Widenius von der schwedischen Firma TcX geschrieben wurde. Aufgrund einer zu schwachen Performanz verfügbarer SQL Server für TcX's große Datenbasis wurde ein eigener SQL Server in Anlehnung an mSQL entwickelt. [DuB06]
MySQL wurde auf Grund seiner Performanz und seiner Einfachheit populär. Das DBMS umfasst einen SQL Server, Programme um auf diesen zuzugreifen, administrative Werkzeuge und eine Programmierschnittstelle, welche es erlaubt eigene Programme zu schreiben [DuB06].

5.2 Implementierung

Die Implementierung verwendet als Grundlage das Empfehlungssystem von [Ihl14], welches für das Wiki des Portals *myKOSMOS* konzipiert und implementiert wurde. Dieses bildet bereits ein gutes Empfehlungssystem, da es inhaltsbasierte, kollaborativ und hybride Methoden für die Empfehlungsberechnung enthält. Auch wenn dieses Programm keinesfalls als vollendet anzusehen ist [Ihl14], sondern mehr als erste Iteration eines hybriden Empfehlungssystems, so war es dennoch hervorragend nutzbar, da es nur geringer Anpassungen bedurfte. Aus diesem Grund wird die reine Beschreibung der Funktionsweise des Programmes nur verkürzt wiedergegeben, da die ausführliche Beschreibung in der Arbeit von [Ihl14] zu finden ist. Veränderungen im Rahmen dieser Arbeit zum ursprünglichen Programm werden gesondert zu jeder einzelnen Phase am Ende angegeben und im nächsten Abschnitt 5.3 bewertet.
Die Software ist als Plugin konzipiert, welches leicht in das laufende Projekt integriert werden kann. Das Plugin ist ein in Java geschriebenes Programm, welches als Java Archive (.jar) dem Portal bereitgestellt wird.

Portale und Portlets mit Hilfe von *Liferay* zu implementieren ist ein sehr komplexes Thema. Es ist zwar möglich einfache Portale und Portlets mit gewissen Grund-Funktionalitäten beispielsweise über die *Liferay Eclipse IDE* zu erstellen, jedoch ist es nötig wesentlich tiefer in die Struktur von *Liferay* einzudringen, wenn ein Portal oder ein Portlet mit fortgeschrittenen Funktionalitäten damit erstellt werden soll. Die Hintergrundstruktur ist nicht eingängig, die Dokumentation muss größtenteils selbst erarbeiten werden und es erfordert einen hohen Grad an Wissen, da die Arbeit mit *Liferay* Kenntnisse von verschiedenen Konzepten und Programmiersprachen (Java, HTML, etc.) voraussetzt.

Die Abbildung 5.1 verschafft einen generellen Überblick über den Ablauf der drei Phasen des Programmes. In der ersten Phase werden von einer Datenbank Datensätze abgefragt, die für die Berechnung der beiden Ansätze des Empfehlungssystems benötigt werden. In der zweiten Phase erfolgen dann die getrennten Berechnungen der Empfehlungen mit dem jeweiligen Ansatz. In der dritten Phase werden die Bewertungen dann aggregiert und die daraus resultierenden finalen Empfehlungen in die Datenbank zurück geschrieben.

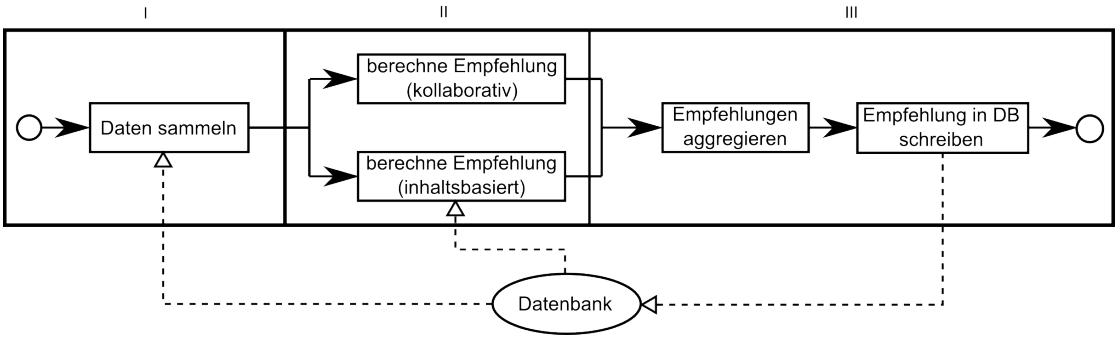

Abbildung 5.1: Allgemeiner Ablauf des Programm (Darstellung nach [Ihl14])

Um das Empfehlungssystem zu starten, ist es nötig eine Instanz der Hauptklasse *RecommendLiferay* zu erzeugen und die Methode `calcRecommendations()` aufzurufen. Diese Methode kapselt weitere Methoden, die sich den einzelnen Phasen zuordnen lassen, siehe Quelltext.

```
public void calcRecommendation() throws Exception {
    getPreferences(); //Phase 1
    getUserList();
    calcRecommendationA(); Phase 2
    calcRecommendationB();
    mergeRecommendations(); //Phase 3
    updateDB();
}
```

Um ein Verständnis für die Funktionsweise des Programms zu vermitteln, wird nachfolgend kurz jede einzelnen Phase dargestellt, dabei wird auf die wesentlichen Punkte der Funktion eingegangen.

Phase 1

Liferays native Datenbankstruktur besitzt eine Tabelle `ratingsentry` in der alle Bewertungen, die im Portal abgegeben werden, gespeichert werden. Die Präferenzen, also die Bewertungen, werden über eine entsprechend formulierte SQL-Abfrage ausgelesen. Weiterhin werden Variablen definiert und die Datenstruktur `PreferenceArray` bereitgestellt. Diese ist eine besondere Datenstruktur von Mahout, in der Präferenzen für einen Benutzer geordnet gespeichert werden können. Sie enthält dabei eine einzelne *UserID* und jeweils ein Array für *ItemIds* und die dazugehörigen Präferenzen. Die Besonderheit an dieser Datenstruktur ist, dass sie besonders speichereffizient ist und zu einem leistungstechnischen Gewinn führt. [OADF11]
Ein solches `PreferenceArray` wird nun für jeden Nutzer befüllt und in die Datenstruktur `FastByIDMap` gelegt. Zu diesem Zweck wird zunächst eine SQL-Anfrage ausgeführt, welche als Ergebnis ein `ResultSet` zurück liefert. Über eine Schleife kann dieses zeilenweise ausgelesen und die `PreferenceArrays` befüllt werden.
Eine zweite SQL-Anfrage dient dazu, die Anzahl der Bewertungen eines jeden Benutzers zu bestimmen, wodurch die notwendige Länge (Speicherplatz) des jeweils erzeugten `PreferenceArrays` akkurat ermittelt und potentielle *OutOfBoundExceptions* vermieden werden können. Ein weiterer Teil der Methode liefert die Datenstruktur an die H auptklasse zurück, welche nun die Präferenzen aller Benutzer beinhaltet.
Dort folgt der Aufruf der Methode `getUserList()`, welche eine Verbindung zum Datenbankserver herstellt und einen `Vector` erzeugt, welcher alle *UserIDs* enthält. Dieser wird für die nachfolgenden Operationen benötigt.

In dieser Phase wurde für die Methode `getPreferences()` lediglich die `classNameId` in der SQL-Abfrage auf den Wert 65432 geändert, um etwaige Kollisionen mit bestehenden Daten in der Tabelle `ratingsentry` zu verhindern. Dazu sei erwähnt, das diese Zahl lediglich ein Platzhalter ist und bei der eigentlichen Umsetzung des Empfehlungssystems entsprechend geändert werden muss. Die `classPK` spiegelt in diesem Falle die Modulnummer wider.

Phase 2

In dieser Phase werden die Empfehlungen für jeden Ansatz getrennt berechnet. Für den kollaborativen Ansatz wird dazu von der Hauptklasse die Methode `calcRecommendationA()` aufgerufen.
Diese Methode liest zunächst die Präferenzen aus **Phase 1** ein, zusätzlich wird ein Objekt einer Klasse erzeugt, die es ermöglicht Ähnlichkeiten zwischen Benutzern zu berechnen, dazu wurde der Euklidische Abstand gewählt. Damit eine Berechnung von Empfehlungen möglich ist, wird noch ein sogenanntes *NearestNeighborhood* Objekt erzeugt, welches wie ein Filter wirkt und die Anzahl der betrachteten Benutzern einschränkt [OADF11]. Dadurch wird eine Empfehlung für einen Nutzer nur mit Hilfe der 10 ihm ähnlichsten Benutzer generiert.
Die Objekte des *NearestNeighborhood* und des *Ähnlichkeitsmaßes* werden an einen Konstruktor übergeben, dieser erzeugt ein Objekt, welches über die Methode `recommend()` für einen einzelnen Benutzer eine gegebene Anzahl von Empfehlungen berechnet. Eine Schleife am Ende des Quelltextes generiert für jeden Nutzer maximal fünf Empfehlungen und speichert die Ergebnisse in den *Vector* `resultsA`, welcher auf den Datentyp `RecommenderData` beschränkt ist. Ein Objekt der Klasse `RecommenderData` speichert genau eine *UserID* und eine *Liste* mit Empfehlungen. Die Empfehlungen haben den Typ `RecommendedItem`, in dem genau eine *ItemId* und der dazugehörige Wert für eine Empfehlung gespeichert wird.

Die Empfehlungen des kollaborativen Ansatzes wurden berechnet und die Ergebnisse gespeichert. Der nächste Schritt ist nun die Berechnung von Empfehlungen mit dem inhaltsbasiertem Ansatz, dazu ruft die Hauptklasse die Methode `calcRecommendationsB()` auf.

Es wird eine weitere Verbindung zum Datenbankserver hergestellt und mit Hilfe der Methode `getTextData()` werden alle Texte mit zugehöriger *ID* ausgelesen und gekapselt im Datentyp `TextData` in einem `Vector` gespeichert.
Dieser wird an die Methode `calcSimilarities()` übergeben, welche die Ähnlichkeiten zwischen allen Texten berechnet und diese in einer `HashMap` abspeichert. Die Berechnung dieser Ähnlichkeiten übernimmt die Klasse `CosineDocumentSimilarity`.

Unter der Nutzung der *Lucene*-Bibliotheken indiziert der Konstruktur der Klasse `CosineDocumentSimilarity` die übergebenen Texte. Zum Einsatz kommt dort ein *Analyzer*, welcher aus den gegebenen Texte *Stopwörter* herausgefiltert. Im weiteren Verfahren ergeben sich zwei intern gespeicherte Vektoren, aus denen mit Hilfe des *Cosinusähnlichkeitsmaßes* ein Wert zwischen -1 und 1 errechnet wird, der den Grad der Ähnlichkeit zweier Texte widerspiegelt. Das Ergebnis der Methode wird für weitere Berechnungen der Methode `CalcRecommendationB()` genutzt.
Die Ähnlichkeiten der Texte müssen mit einer geeigneten Strategie verrechnet werden. Die Literatur bietet dafür wenig Material, da es keine pauschalen Lösungen für inhaltsbasierte Empfehlungssysteme gibt. Sie hängt maßgeblich davon ab, wie Ähnlichkeit der Objekte definiert wird und welche Methoden dafür verwendet werden, um eine Ähnlichkeit festzustellen. Abhilfe hat hier die Übernahme einer Formel gebracht, die eigentlich bei kollaborativen Empfehlungssystemen verwendet werden. Die Ähnlichkeit zwischen zwei Nutzern wird mit der Ähnlichkeit zwischen zwei Texten i und i' in der aus Abschnitt 2.1.5 bekannten Formel ersetzt. Sodass

$$r_{c,s} = k \sum_{c' \in \hat{C}} sim(i, i') \times r_{c',i}$$

mit $k = 1/\sum_{i' \in \hat{I}} |sim(i, i')|$.
Die Bewertung für ein unbewertetes **item** i des **users** c ergibt sich aus der Summe aller Ähnlichkeiten zu bereits bewerteten Items i' aus der Menge I multipliziert mit der Bewertung für i'. Zur Normalisierung wird durch die Summe der Beträge der Ähnlichkeiten geteilt. Diese Rechnung wird für alle unbewerteten **items** aller Nutzer berechnet. Das Ergebnis wird in der Datenstruktur `resultsB` abgelegt.

In dieser Phase wurde der SQL-Befehl für die Abfrage der Vorlesungsinhalte entsprechend angepasst. Dabei wurde eine neue Datenstruktur namens `vl_data` eingeführt, um Vorlesungen entsprechend effizient abspeichern zu können. Diese Struktur wurde prototypisch für diese Arbeit erstellt und beinhaltet eine *fortlaufende Nummerierung*, die *Modulnummer* und den *Namen* der Vorlesung sowie den *Vorlesungsinhalt* als Text.

Phase 3

Die Ergebnisse der beiden Verfahren liegen in getrennten Datenstrukturen vor, in der letzten Phase werden beide Strukturen zu einer Einzigen vereint.
Dabei wurde eine Strategie verwendet, sodass die Ergebnisse aus dem inhaltsbasierten Ansatz auf jeden Fall erhalten bleiben. Liegen Empfehlungen für einen Text aus beiden Verfahren vor, so wird der Mittelwert gebildet. Im Laufe der Programmierung hat sich herausgestellt, dass bei ungünstiger Konstellation der Präferenzen keine Empfehlungen von *Mahout* für einige Benutzer generiert werden. Um dennoch Empfehlungen zu erhalten, wurde dies mit den Bewertungen aus dem inhaltsbasierten Ansatz abgefangen. Ein Schleifenkonstrukt löst das Vermischen der Ergebnisse. [Ihl14]
Die Ergebnisse werden in der Datenstruktur `resultsFinal` abgelegt. Im letzten Schritt von **Phase 3** wird die Methode `updateDB()` aufgerufen, diese stellt eine Verbindung zum Datenbankserver her und schreibt die Ergebnisse der Berechnungen in die Tabelle `recommendations`. Ist bereits eine Tabelle mit alten Ergebnissen vorhanden, so werden diese vorher gelöscht.
Über den Wert der *classPK*, also die Modulnummer in diesem Beispiel, können die **items** eindeutig identifiziert werden.

In der letzten Phase wurde in der Methode `updateDB()` eine kleine Form der Effizienzoptimierung durchgeführt. In der originalen Fassung der Methode wurde für jede neue Empfehlungsberechnung die Tabelle `recommendations` jeweils komplett gelöscht und anschließend neu erstellt. Dies ist aus der technischen Sicht einer Datenbank ungünstig, da es eine relativ hohe Belastung in der Datenbank verursacht, zudem ist die Tabelle `recommendations` lediglich ein Tabelle mit temporären Werten. Deswegen wurde das gesamte Konstrukt des Löschens und des Neuerstellens der Tabelle mit dem SQL-Befehl `TRUNCATE TABLE recommendations` ersetzt. Dies setzt nun natürlich voraus, dass bereits eine solche Tabelle existiert. Dafür kann diese aber immer wieder verwendet werden, da nur die Werte der Tabelle selbst gelöscht werden, die Struktur und Definition aber für zukünftiges Beschreiben vorhanden bleiben [sql15]. Deswegen muss bei der Konzeption der Datenbank eine solche Tabelle berücksichtigt werden.
Auch wurde eine Fehlerbehandlung eingeführt, die bei undefinierten Werten von Empfehlungen einen Default-Wert liefert.

5.3 Bewertung der Implementierung

Dieser Abschnitt wird die Bewertung der Implementierung enthalten. Im ersten Teil wird das bereits bestehende Programm aus der Arbeit von [Ihl14] kurz analysiert und bewertet. Dabei werden im Übergang zur Bewertung des erweiterten Programms aus dieser Arbeit alle dafür notwendigen Änderungen genannt und bewertet.

Ein erster Test mit mutmaßlich ähnlichen Daten wie in der Arbeit von [Ihl14] lieferte ebenfalls ähnliche Ergebnisse. Hierzu muss beachtet werden, dass aufgrund der Arbeit von [Ihl14] zwar die Bewertungen rekonstruiert werden konnten, wodurch der kollaborative Teil des RCS in der erste Iteration zu den gleichen Ergebnissen kam wie in besagter Arbeit, die zweite Iteration in der die Empfehlungen gemischt wurden liefert ein leicht anderes Bild, vgl. Tabelle 5.1. Da die Texte der Inhaltsbasierten Methode nicht mehr vorhanden waren, wurden generische Texte genutzt, um ein vergleichbares Ergebnis zu erhalten. Hierbei sind wie in der Arbeit von [Ihl14] die 10x-Texte dem Bereich Sport, die 20x-Texte dem Informatik Bereich und die 30x-Texte der Physik zuzuordnen.

	Arbeit von [Ihl14]			Diese Arbeit		
Nutzer	Kollab.	Inhaltsb.	Hybrid	Kollab.	Inhaltsb.	Hybrid
1	202, 201, 203	204, 201, 202	202, 201, 203	202, 201, 203	104	104
2	102, 104, 103	204, 301, 304	204, 301, 304	204, 102, 104	204, 103, 104	204, 102, 104
3	101, 104	104, 201, 304	101, 201, 304	101, 304, 303	304, 303, 104	101, 303, 304
4	201, 202, 101	101, 303, 202	202, 201, 101	201, 202, 101	202, 201, 101	202, 201, 101
5	203	203, 204, [!sic]402	203, 204, 302	204, 203	203, 204	203, 204

Tabelle 5.1: Vergleich der Top3 Empfehlungen des RCS der Tabelle `recommendations`

Wie in der Tabelle 5.1 ersichtlich ist, stimmen die Ergebnisse zu etwa 66% überein. Nur gleiche Einträge wurden dazu gewertet, das Minimum an Einträgen jeder Zelle ist das absolute Maximum, was pro Zelle gezählt wird.

Die Abweichung ist vorrangig dadurch zu erklären, dass insgesamt das tatsächliche Ergebnis in der Tabelle berücksichtigt wurde. Beim kollaborativen Teil gibt es im Programm vor der Zusammenführung der Datenstrukturen einen Status der exakt dem im linken Teile der Tabelle 5.1 unter "Kollaborativ" dargestellten Empfehlungen entspricht. Die in die Datenbanktabelle `recommendations` zurückgeschriebenen Werte entsprechen am Ende nur noch zu ca. 83% den Werten aus der Arbeit von [Ihl14]. Dies liegt an der Art und Weise der Zusammenführung der jeweiligen Ergebnisse der einzelnen Methoden im Programm, bei der immer kollaborativ wie inhaltsbasiert berechnet werden und bei der Zusammenführung immer der inhaltbasierte Teil bestehen bleibt. Zudem ergibt sich im inhaltsbasierten Ansatz eine Abweichung, da hier nun einerseits andere Texte verwendet wurden, was die Vergleichbarkeit erheblich erschwert, andererseits lieferte die Ähnlichkeitsbestimmung mittels *Apache Lucene* auch teilweise `NaN`-Werte, welche dann für die bessere Handhabbarkeit auf einen Defaultwert (-1.0) gesetzt wurden, um so das Programm lauffähig zu halten. Dadurch kann sich natürlich ein großer Teil an "Nicht-Empfehlungen" ergeben, dies gilt es durch gezielte Arbeit an der inhaltsbasierten Komponente zu verbessern. Dies soll allerdings nicht mehr Teil dieser Arbeit sein, da der zeitliche Rahmen durch das Konzipieren des Empfehlungssystems und einer prototypischen Implementierung bereits ausgeschöpft ist. Zudem erfordert der qualifizierte Umgang mit *Apache Lucene* eine lange Einarbeitungszeit.

Nach diesem ersten Funktionstest wird nun auf die notwendigen Änderungen im Programm selber eingegangen, die im Rahmen dieser Arbeit getätigt wurden.

Das Programm wurde im Rahmen dieser Arbeit auf ein Empfehlungssystem für Vorlesungen, genauer von Wahlpflichtmodulen, umgeschrieben, welches wahlweise, wie das originale Programm auch den inhaltsbasierten, den kollaborativen wie auch den hybriden Ansatz zur Berechnung von Empfehlungen anwenden kann.
Für die Befüllung der Datenbank dient der Wahlpflichtmodulkatalog für die Wirtschaftsinformatiker der Universität Rostock [ur.15a]. Auf eine Unterscheidung zwischen den weiterführenden Studienrichtungen *Informationssysteme* sowie *Business Informatics* wurde aus zeitlichen Gründen bei der Implementierung verzichtet. Für den Test werden aus jedem Wahlpflichtkatalog jeweils zwei Vorlesungen aus dem Wintersemester sowie dem Sommersemester betrachtet, um so eine gute Vergleichbarkeit zu erhalten.

Als Testwerte dienen einerseits Bewertungen der Vorlesungen, diese werden in der zentralen Bewertungstabelle `ratingsentry` von *Liferay* gespeichert. Andererseits werden, so vorhanden, die Beschreibung des Inhaltes (Lerninhalt) der Vorlesungen aus der *LSF-Suche* der Universität Rostock verwendet [ur.15b]. Die *classPK* entspricht in diesem Testfall jeweils der zugehörigen Modulnummer, um eine effiziente Zuordnung zu gewährleisten. Die Einträge in die Datenbanktabellen wurden mittels SQL-Skripten erstellt, diese sind auf der CD dieser Arbeit unter `Prototyp/SQL-Scripts` zu finden.
Als Test-Vorlesungen wurden die folgenden Module gewählt:

- Funktionale Programmierung
- Logik und Berechenbarkeit
- Computergraphik
- Künstliche Intelligenz (Modul: Smart Computing)
- Dienstleistungsmanagement (Unternehmensführung)
- Einführung in die Wirtschaftsprüfung
- Strategisches Marketing (Marketing)
- Einführung in die betriebswirtschaftliche Steuerlehre
- Vertiefungsstufe Fachkommunikation Wirtschaftswissenschaften Modul 2 C1.1.2
- Vertiefungsstufe Fachkommunikation Ingenieurwissenschaften Modul 3 C1.2
- Vertiefungsstufe Fachkommunikation Wirtschaftswissenschaften Modul 1 C1.1.1
- Vertiefungsstufe Fachkommunikation Wirtschaftswissenschaften Modul 3 C1.2

Phase 1

In der ersten Phase des Programms wurde der SQL-Abfrage-String der Klasse MySQLCon() für die Bedürfnisse dieser Arbeit angepasst. Die Struktur wurde beibehalten, lediglich die *classNameId* wurde auf den Wert "65432" gesetzt um Kollisionen mit eventuell beim Test bestehenden anderen Einträgen in der Tabelle `ratingsentry` zu vermeiden.

Dieser Wert ist lediglich ein Testwert. Für den tatsächlichen Einsatz ist es notwendig zu überprüfen, ob diese *classNameId* schon anderweitig in Liferay vergeben wurde. Zudem sollte eine kollisionsfreie interne Nummerierung gefunden werden.

Phase 2

In der Phase 2 wurde in der Methode `getTextData()` der SQL-Befehl für die selbsterstellte Datenstruktur der Vorlesung angepasst. Dies hatte den Hintergrund, dass *Liferay* keine Datenstruktur anbietet, die eine Vorlesung zufriedenstellend aufnehmen könnte, weswegen eine eigene kleine Struktur entworfen wurde. Diese Struktur namens `vl_data` umfasst eine *inkrementelle Nummerierung*, die *Modulnummer*, den *Namen* für eine effizientere Zuordnung für die Präsentation sowie natürlich die *Beschreibung des Vorlesungsinhaltes*, so vorhanden. Auf andere Attribute wie den *Dozenten* oder die *Uhrzeit* wurde in diesem Prototyp verzichtet, diese sind natürlich im fertigen Empfehlungssystem für die Präsentation erforderlich.

Phase 3

In der letzten Phase wurde die Methode `updateDB()` dahingehend verändert, dass statt eines Löschens und Neuerstellens der Tabelle `recommendations` diese nun nur noch durch den SQL-Befehl `TRUNCATE TABLE` geleert wird. Dies hatte den Zweck, die Last auf die Datenbank zu minimieren, da die Tabelle `recommendations` zwar nur temporäre Werte für Empfehlungen enthält, diese aber nicht für jede Empfehlungsberechnung gelöscht und dann neu erstellt werden muss. Zudem bleibt die Struktur und Definition der Tabelle durch `TRUNCATE TABLE` erhalten, wodurch die Tabelle nur ein einziges Mal, dafür aber zwingend, bei der Initialisierung der Datenbank erstellt werden muss.
Bei der Berechnung des inhaltsbasierten Ansatzes traten vereinzelt Fehler in Form eines `NaN`-Wertes auf. Diese mussten abgefangen werden, da inhaltsbasierte Empfehlungen in jedem Falle erhalten bleiben und deswegen beim Zurückschreiben in die Datenbank Fehler und den Abbruch des Prozesses verursachen. Deswegen wurde entschieden, einen Default-Wert von -1.0 für `NaNWerte` einzusetzen, um so die Lauffähigkeit des Programms zu sichern und diese Werte kenntlich zu machen, damit diese Fehler schneller behoben werden können.

Die Ergebnisse des veränderten Empfehlungssystems werden in Tabelle 5.2 dargestellt, die Nummern entsprechen den jeweiligen Modulnummer der Vorlesungen aus dem *LSF* der Universität Rostock, welche weiter oben zu finden sind.

Nutzer	Kollaborativ	Inhaltsbasiert	Hybrid
1	50121, 50162, 54518	54518, 90038, 900037	50162, 54518, 900037
2	23507, 900037, 90015	23507, 900037, 90015	23507, 900037, 90015
3	23013, 54518, 90039	54518, 90015, 90039	23013, 54518, 90039
4	23013, 50144, 90038	50144, 54518, 90038	23013, 50144, 90038
5	90039, 90015, 900037	90039, 90015, 900037	90039, 90015, 900037

Tabelle 5.2: Top3 Empfehlungen des RCS der Tabelle `recommendations`

Nutzer 1 hat eine Vorliebe für die Informatik, er bekommt im kollaborativen Ansatz vorrangig Vorlesungen aus dem Bereich der Wirtschaft empfohlen, dies ist eine direkte Folge der Bewertungen der Nutzer 4 und 5, da diese sowohl die Informatik als auch die Wirtschaft als "gut'"bewerteten, sowie der begrenzten Testdaten. Bei Nutzer 2 verhält sich ähnlich, er bewerte die Wirtschaftsvorlesungen als gut und die der Informatik schlecht, so erhielt er vorrangig Empfehlungen für Sprachen. Nutzer 3 zeigt ein sehr ausgewogenes Bild, er bewertete die Informatik und die Sprachen gut und bekam so Empfehlungen aus allen 3 Bereichen, ebenso wie Nutzer 4, obwohl dieser tendenziell die Informatik schlechter bewertete. Nutzer 5 bewertete sowohl die Informatik als auch die Wirtschaft positiv, dadurch bekam er vorrangig Empfehlungen für Sprachen.
Der inhaltsbasierte Ansatz zeigt ähnliche Ergebnisse wie der Kollaborative für die Nutzer 2 und 5. Nutzer 1 erhält nun mehr Empfehlungen aus dem Bereich der Sprachen, ebenso wie Nutzer 3, der die Empfehlungen aus der Wirtschaft beibehält. Zu erwarten gewesen wäre, dass Nutzer 1 Empfehlungen aus dem Informatikbereich erhält. Nutzer 4 zeigt ein leicht anderes Bild, da bei ihm die Informatikvorlesungen zugunsten der Wirtschaftsvorlesungen vertauscht wurden. Diese Ergebnisse dürften den stellenweise sehr kurzen oder gar nicht vorhandene Beschreibungen der Vorlesungsinhalte geschuldet sein.
Der hybride Ansatz verhält sich wie erwartet, da die Ergebnisse jeweils gemischt werden, sind die hybriden Empfehlungen eine Kombination aus den beiden anderen Ansätzen mit einer Tendenz zu denen des inhaltsbasierten Ansatzes.

Trotz vereinzelter Ausreißer bei den Empfehlungen ist das Ergebnis im Allgemeinen als zufriedenstellend anzusehen.

Laufzeiten

In der Tabelle 5.3 wurden die jeweiligen Laufzeiten der Ansätze der Empfehlungsberechnung im Programm zusammengefasst. Die Zeitangaben beziehen sich auf das Ende des jeweiligen Vorganges.

Vorgang	Zeit in s		
	Kollaborativ	Inhaltsbasiert	Hybrid
Daten sammeln	0.372	0.351	0.364
Empfehlung kollaborativ	0.415	0.391	0.407
Empfehlung inhaltsbasiert	1.060	1.017	1.015
Empfehlungen mischen	1.061	1.017	1.016
Empfehlungen in DB schreiben	1.266	1.236	1.199

Tabelle 5.3: Laufzeiten des Programms je Ansatz (eigene Darstellung)

Wie aus der Tabelle 5.3 ersichtlich ist, liegen die Laufzeiten der einzelnen Ansätze des Programms sehr dicht beeinander und enden spätestens nach etwa 1,2s. Etwa 0,5s davon entfallen auf Datenbankoperationen und deren Kommunikation mit dem Programm, davon sind etwa 0,3s der erste Lesevorgang, wohingegen der letzte Schreibvorgang in die DB nur 0,2s ausmachen. Der kollaborative Ansatz erzeugte sehr schnell Ergebnisse, in lediglich etwa 0,04s waren die Berechnungen abgeschlossen. Der inhaltsbasierte Ansatz verbraucht jeweils in etwa 0,6s, dies könnte darauf hindeuten, dass bei größeren Texten und Textmengen Performanzeinbußen zu verzeichnen sein könnten, dies sollte genau getestet und entsprechend optimiert werden. Hier könnte das Design-Pattern *Observer* Abhilfe schaffen, da so nur Veränderungen berechnet werden würden.
Die Messungen wurden auf einem *AMD Phenom II X6 1100T BE* mit Standardtaktung durchgeführt, als Datenspeicher diente eine *Samsung 850 PRO* SSD. Zum Stand dieser Arbeit ist die CPU leistungstechnisch im Mittelfeld (verglichen mit anderen AMD CPUs im oberen Mittelfeld) [amd15], die SSD in der Leistungsspitze anzusiedeln [sam15].

6 Zusammenfassung & Ausblick

Dieses Kapitel soll den Abschluss der Arbeit durch eine Zusammenfassung der vorangegangenen Kapitel und deren Inhalten darstellen. Abschließend wird ein kurzer Ausblick in die Zukunft des Portals *myKOSMOS* und dieses Konzeptes gegeben.

6.1 Zusammenfassung

In dieser Arbeit wurden zuerst die Grundlagen gelegt, die für ein weiteres Verständnis des Konzeptes, welches im Rahmen dieser Arbeit erarbeitet wurde, notwendig sind. Es wurde in die Thematik der Empfehlungssysteme eingeführt, ebenso erfolgten Einführungen zum Thema Informationslogistik, bedarfsorientierte Informationsbereitstellung, der Thematik der Portale und der Portalsoftware *Liferay*. Abschließend erfolgte eine kleine Übersicht über bereits entwickelte oder sich in der Entwicklung befindliche Portlets für das Portal *myKOSMOS*.
Daran anschließend wurde in Kapitel 3 kurz das Vorgehen für das Erstellen des Konzeptes eines Empfehlungssystems für ein Portal erklärt, damit der Leser nachvollziehen konnte, in welcher Art und Weise das vorliegende Portal untersucht wurde, um dieses Konzept zu erstellen und was Bestandteil dieses Konzeptes sein würde.
Danach wurde in Abschnitt 4.1 mit der Konzeption aus Sicht des Einsatzzweckes begonnen. Dies lieferte bereits große, wenn auch stellenweise ungenaue Einsatzgebiete für ein solches Empfehlungssystem. Anschließend wurde aus Sicht der Prozessorientierung in Abschnitt 4.2 feiner in diese Gebiete aus dem ersten Abschnitt eingegangen, um so empfehlungswürdige **items** für **user** zu finden. Dies gelang am Anfang recht schwierig, wurde aber schnell leichter und führte deshalb schnell zu einer großen Menge an Strukturen, die alle in das Konzept zu integrieren den Rahmen dieser Bachelor-Arbeit und auch das zeitliche Limit bei der Bearbeitung gesprengt hätten.

Deswegen wurde entschieden, ab einer bestimmten Tiefe nicht weiter in die Einsatzgebiete einzudringen und nur einen kleinen Überblick über die jeweilige Ebene zu liefern. Auf der technischen Ebene, welche sich in Abschnitt 4.3 anschloss, wurde zuerst die Datenebene des Portals untersucht. Auch hierbei stellte sich heraus, das diese Ebene sehr schnell umfangreicher wird, je tiefer man bei der Bearbeitung eindringt. Deswegen wurde auch hier beschlossen, sich nur auf die wichtigsten Komponenten bei einer *LifeRay*-Neuinstallation zu beschränken. Ebenso wurde bei der Untersuchung der für das Portal bereits im Rahmen von wissenschaftlichen und studentischen Arbeiten entwickelten Portlets verfahren. Hierbei wurde festgestellt, dass das *Support2-Portlet* aus der Arbeit von [Ack14] bereits einen hervorragenden Ansatz für eine Beobachtungskomponente enthält. Ebenso enthalten das *Request/Wegtam-Portlet*, das *Skype-Portlet* von [DVWG14], das sich zu dem Zeitpunkt dieser Arbeit noch in der Entwicklung befindliche *StudIP-Portlet* von [Spe15] und das *Google Drive-Portlet* von [Wei14] auch Ansätze, die für die Zukunft des Portales von entscheidender Bedeutung sind. Anschließend wurde auf der technischen Ebene auf die Bedeutung der Hooks für die Bobachtungskomponente eingegangen. Darauf folgend wurde ein Bezug zu den empfehlungswürdigen **items** aus Abschnitt 4.2 gebildet und für diese **items** für ein Empfehlungssystem wichtige Daten definiert.
In Abschnitt 4.4 wurden Erweiterungen vorgestellt, welche die zukünftige Arbeit eines Empfehlungssystem im Portal positiv unterstützen können, sei es durch die Bereitstellung zusätzlicher Daten für die Empfehlungsberechnung oder einer Möglichkeit Empfehlungen zielgerichtet an den potentiellen Empfänger, den **user**, zu bringen. Die wünschenswerten Erweiterungen wurden so genau wie möglich beschrieben, ebenso ihr potentieller Beitrag für die Arbeit des Empfehlungssystems und die benötigten Datenstrukturen für jede Erweiterung, damit das Empfehlungssystem zukünftig effizient darauf arbeiten kann.
Anschließend wurde in Abschnitt 4.5 auf die Bedeutung des Datenschutzes eingegangen. Besonderes Augenmerk lag dabei auf der Thematik der *k-Anonymität*, da diese für Empfehlungssystem eine wichtige Rolle spielt. Kurz zusammengefasst geht es bei der *k-Anonymität* um den Schutz von sensiblen Daten durch Aggregation zu Gruppen, die sich gleiche, nicht-sensible Daten teilen. Diese bietet, wie erarbeitet wurde, dennoch keinen vollkommenen Schutz, da durch die Ausnutzung verschiedener Umstände, von denen die zwei populärsten beschrieben wurden, eine eindeutige Identifizierung von Personen möglich ist. Um diese Schwachstellen zu umgehen, wurden diese im Konzept aufgezeigt und, so vorhanden, Lösungsansätze aufgezeigt.

Der Abschnitt 4.6 befasste sich mit der allgemeinen Präsentation der Empfehlungen im Portal. Dazu wurde zuallererst die Notwendigkeit der Visualisierung der Empfehlung aus Sicht des Nutzers diskutiert. Anschließend wurden verschiedene Beispiele von Portalen und Websites betrachtet und deren Präsentation von Empfehlungen untersucht und evaluiert, damit bei der Umsetzung des Empfehlungssystems positive Beispiele als Vorlage zur Verfügung stehen.
Der vorletzte Abschnitt 4.7 beschäftigte sich mit der Notwendigkeit, die Qualität von Empfehlungen zu überprüfen. Im Rahmen dieses Abschnittes wurden verschiedene Möglichkeiten der Überprüfung diskutiert, ebenso wurde auch die Visualisierung diskutiert.
Abschließend für das Kapitel 4 wurde das gesamte Konzept in Abschnitt 4.8 einer kritischen Bewertung unterzogen.
Für das Kapitel 4 ist zu beachten, dass das erarbeitete Konzept für ein Empfehlungssystem in einem Portal nicht als fertiges Konzept, sondern als eine erste Iteration eines solchen zu verstehen ist. Um ein quasi allumfassendes Konzept für ein Empfehlungssystem zu erstellen, wäre wesentlich mehr Zeit und tiefergreifende Kenntnisse der Vorgänge der Universität Rostock und der Portal-Software notwendig gewesen. Auch hätte sich eine umfassendere Konzeption negativ auf die Lesbarkeit dieser Arbeit ausgewirkt, da das Konzept in der jetzigen Form bereits rund 65 Seiten einnimmt.
Kapitel 5 widmete sich der Implementierung eines kleinen Teils des in Kapitel 4 erarbeiteten Konzeptes, genauer um die Empfehlung von Wahlpflichtfächern.
Dazu wurde im ersten Abschnitt 5.1 die für die Implementierung eines Teils des Konzeptes verwendete Software kurz vorgestellt und erläutert.
Der Abschnitt 5.2 widmete sich kurz der Beschreibung der Funktionsweise des Programms von [Ihl14]. Die eigenen Arbeiten an der Software wurden in Abschnitt 5.2 an den entsprechenden Stellen beschrieben. In Abschnitt 5.3 wurden diese bewertet. Dabei stellte sich heraus, dass das Empfehlungssystem von [Ihl14] flexibel konzipiert ist, sodass es sehr schnell auf die jeweiligen Bedürfnisse angepasst werden konnte.

6.2 Ausblick

Da diese Arbeit nicht als fertiges Konzept, sondern als eine erste Iteration eines solchen zu verstehen ist, ist notwendig, dieses Konzept in Zukunft genauer zu überwachen. Dabei ist zu überprüfen, welche Teile schon umgesetzt wurden und welche ggf. nicht, so wie konzipiert, umsetzbar sind.

Auch welche neuen technologischen Entwicklungen es innerhalb der nächsten Jahre geben wird, wie diese in einer neuen Iteration dieses Konzeptes berücksichtigt werden können und was in diesem Konzept noch nicht berücksichtigt worden ist, muss überprüft werden. All diese Punkte sind dennoch nur wenige Beispiele, wie dieses Konzept auf Aktualität überprüft werden kann, denn dies kann und muss regelmäßig geschehen, um den sich ändernden Anforderungen gerecht zu werden.
Ebenso ist es wichtig, da in dieser Arbeit nur die Empfehlungskomponente konzipiert wurde, ebenfalls eine Beobachterkomponente zu konzipieren und zu implementieren, um das Empfehlungssystem so gut wie möglich zu unterstützen. In diesem Punkt leistete die Arbeit von [Ack14] schon eine hervorragende Vorarbeit, da in jener Arbeit bereits ein Teil einer Beobachtungskomponente konzipiert und auch implementiert wurde. Auf Basis dieser Arbeit und der dabei entstandenen Software sollte dringend weiter gearbeitet werden, auch unter dem Gesichtspunkt, das man sich mit so einer Lösung vor eventuellen Inkompatibilitäten schützen kann. Auch wurde in jener Arbeit ein Teil eines Empfehlungssystemes entworfen, welches mit Hilfe von Regeln Auswertungen der durch die Beobachterkomponente erstellten Daten in der Datenbank ermöglicht.
Auch wurde im Rahmen dieser Arbeit ein Empfehlungssystem-Prototyp von [Ihl14] entsprechend erweitert, angepasst und optimiert, um den Bedürfnissen der Implementierung eines Teils des in dieser Arbeit entworfenen Konzeptes zu genügen. An dieser Stelle ist wichtig, dass dieser Ansatz der Empfehlungsberechnung mit dem im vorherigen Absatz erwähnten regelbasierten Empfehlungssystem verglichen wird. Dies muss getan werden, um herauszufinden, ob beide Ansätze parallel auf verschiedene Gebieten, oder nur einer von beiden Ansätze im Portal angewendet werden soll. Entscheidend ist jedoch, dass wenn beide Ansätze (oder mehr), in einer Empfehlungskomponente angewendet werden, diese Komponente nach außen eine Einheit bildet.

Es ist noch viel Arbeit nötig, um das Portal effizient zu gestalten, sei es bei der Benutzeroberfläche oder in den darunter agierenden Portlets oder in der Datenschicht des Portals. In den Ebenen auf denen die Empfehlungskomponente und die Beobachtungskomponente arbeiten werden, muss noch viel entwickelt werden, damit das Konzept des Empfehlungssystems dieser Arbeit effizient und vernünftig umgesetzt werden kann.

Literaturverzeichnis

[Ack14] Samuel Ackermann. Bildung von Nutzerprofilen aus dynamischen Nutzungsdaten im Lernportal des KOSMOS Projektes. Master's thesis, Universität Rostock, September 2014.

[Ahn08] Hyung Jun Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*, 178(1):37–51, 2008.

[amd15] Prozessorvergleich - Die große Prozessor Rangliste als CPU Vergleich. `http://www.prozessorvergleich.org/`, 2015. 24.01.2015 16:00.

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[BG10] Martin Böhringer and Martin Gaedke. Ubiquitous microblogging: A flow-based front-end for information logistics. In *Business Information Systems Workshops*, pages 158–167. Springer, 2010.

[Bur02] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[BWZV07] Freimut Bodendorf, Stefan Winkler, Roland Zimmermann, and Bernd Vogele. Agent-based information logistics in planning processes. In *Systems, 2007. ICONS'07. Second International Conference on*, pages 54–54. IEEE, 2007.

[can15] Canvas Fingerprinting - What You Need to Know About the Sneakiest New Online Tracking Tool. `http://gizmodo.com/what-you-need-to-know-`

about-the-sneakiest-new-online-tr-1608455771, 2015. 26.01.2015 14:20.

[DLP03] Wolfgang Deiters, Thorsten Löffeler, and Stefan Pfennigschmidt. The information logistics approach toward user demand-driven information supply. In *Cross-media service delivery*, pages 37–48. Springer, 2003.

[DuB06] Paul DuBois. *MySQL cookbook*. " O'Reilly Media, Inc.", 2006.

[DVWG14] Huy Duc Do, Georg Voss, Dorian Wojda, and Luis Gil. Abschlussbericht für das Projekt Port2Port, August 2014. Projektarbeit.

[Eck10] Peter Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies*, pages 1–18. Springer, 2010.

[ecl14a] Eclipse - About Eclipse. www.eclipse.org/org/, 2014. 04.12.2014 16:30.

[ecl14b] Eclipse - Versions. http://www.eclipse.org/downloads/, 2014. 04.12.2014 16:30.

[Fel10] Jerry Felten. *Empfehlungssysteme im E-Commerce*. GRIN Verlag, 2010.

[HA07] Anne Håkansson and Mats Apelkrans. Enterprise systems configuration as an information logistics process: A case study. In *Proceedings of 9th International Conference on Enterprise Information Systems, ICEIS-2007*, pages 212–220, 2007.

[HBVBK94] Wolfgang Hesse, Georg Barkow, Hubert Von Braun, and H-B Kittlaus. Terminologie der software-technik. ein begriffssystern fuer die analyse und modellierung von anwendungssystemen. teil 1: Begriffssystematik und grundbegriffe. *Informatik Spektrum*, 17:39–39, 1994.

[HK07] Stefanie Hohfeld and Melanie Kwiatkowski. Empfehlungssysteme aus informationswissenschaftlicher Sicht – State of the Art. *Information Wissenschaft und Praxis*, 58(5):265, 2007.

[Ihl14] Marc Ihlenfeld. Einbindung einer (Gruppen-)Empfehlungssystemkomponente für ein Lernportal in der wissenschaftlichen Weiterbildung basierend auf Liferay. Bachelor-Arbeit, Februar 2014.

[jol14a] Jolla - Smartphone. https://jolla.com/jolla/, 2014. 16.12.2014 20:30.

[jol14b] Jolla - The Other Half. https://jolla.com/the-other-half/, 2014. 16.12.2014 20:30.

[jsr15] The Java Community Process(SM) Program - JSRs Java Specification Requests - detail JSR 286. https://www.jcp.org/en/jsr/detail?id=286, 2015. 15.01.2015 16:40.

[K+05] Helmut Krcmar et al. *Informationsmanagement*, volume 5. Springer, 2005.

[Kla09] André Klahold. *Empfehlungssysteme: Recommender Systems-Grundlagen, Konzepte und Lösungen*. Springer DE, 2009.

[kos14] Über KOSMOS. www.kosmos.uni-rostock.de/ueber-kosmos/, 2014. 22.11.2014 15:22.

[Lau11] Dietlinde Lau. *Algebra Und Diskrete Mathematik 1*. Springer, 2011.

[Leh14] Franz Lehner. *Wissensmanagement: Grundlagen, Methoden und technische Unterstützung*. Carl Hanser Verlag GmbH Co KG, 2014.

[lif14] Liferay - Specifications. http://www.liferay.com/de/products/liferay-portal/tech-specs/, 2014. 26.11.2014 13:00.

[lif15a] Liferay - LifeRay Sync. https://www.liferay.com/de/products/liferay-sync/features, 2015. 08.01.2015 16.05.

[lif15b] Liferay - User Guide 6.2 - User Groups. https://www.liferay.com/de/documentation/liferay-portal/6.2/user-guide/-/ai/user-groups-liferay-portal-6-2-user-guide-16-en, 2015. 01.01.2015 22:20.

[lif15c] Liferay - Wiki - Members Portlet. https://www.liferay.com/de/community/wiki/-/wiki/Main/Members+Portlet, 2015. 01.01.2015 21:10.

[lif15d] Liferay - Wiki - Message Boards Portlet. https://www.liferay.com/de/community/wiki/-/wiki/Main/Message+Boards+Portlet, 2015. 22.01.2015 12.50.

[lif15e] Liferay - Wiki - Portlets. https://www.liferay.com/de/community/wiki/-/wiki/Main/Portlets, 2015. 01.01.2015 21:10.

[lif15f] Liferay - Workflow with Kaleo. https://www.liferay.com/de/documentation/liferay-portal/6.0/administration/-/ai/workflow-with-kal-2, 2015. 22.01.2015 12:40.

[lif15g] Pro Liferay - Liferay Quartz Scheduler. http://proliferay.com/liferay-quartz-scheduler/, 2015. 22.01.2015 16:45.

[LLSS06] Tatiana Levashova, Magnus Lundqvist, Kurt Sandkuhl, and Alexander Smirnov. Context-based modelling of information demand: Approaches from information logistics and decision support. 2006.

[LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, volume 7, pages 106–115, 2007.

[LWS08] Franz Lehner, Stephan Wildner, and Michael Scholz. *Wirtschaftsinformatik*, volume 2. 2008.

[LY07] Zhen Li and Xiaojun Ye. Privacy protection on multiple sensitive attributes. In *Information and Communications Security*, pages 141–152. Springer, 2007.

[MAGM13] Bernd Michelberger, Ralph-Josef Andris, Hasan Girit, and Bela Mutschler. A literature survey on information logistics. In *Business Information Systems*, pages 138–150. Springer Berlin Heidelberg, 2013.

[Mar12] André Marchand. *Empfehlungssysteme für Gruppen: Entscheidungsunterstützung für den gemeinsamen Konsum hedonischer Produkte*, volume 4. BoD–Books on Demand, 2012.

[MC07] Nikos Manouselis and Constantina Costopoulou. Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10(4):415–441, 2007.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity.

ACM Transactions on Knowledge Discovery from Data (TKDD), 1(1):3, 2007.

[MMR12] Bernd Michelberger, Bela Mutschler, and Manfred Reichert. Process-oriented information logistics: Aligning enterprise information with business processes. In *Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International*, pages 21–30. IEEE, 2012.

[Ner08] Friedemann W Nerdinger. *Grundlagen des Verhaltens in Organisationen*. W. Kohlhammer Verlag, 2008.

[OADF11] Sean Owen, Robin Anil, Ted Dunning, and Ellen Friedman. *Mahout in action*. Manning, 2011.

[PO09] Wiebke Putz-Osterloh. *Sozialpsychologie der Schule und Familie*. Universität Bayreuth, Wintersemester 2008/2009.

[por15] Skype Chat & IM - Liferay Apps - Rivet Logic Corp. `http://wiki.rivetlogic.com/pages/viewpage.action?pageId=27656493`, 2015. 02.01.2015 15:45.

[pyd14] PyDev - Python IDE for Eclipse. `http://pydev.org/`, 2014. 04.12.2014 16:35.

[sam15] SSD-Rangliste - ComputerBase. `http://www.computerbase.de/thema/ssd/rangliste/`, 2015. 24.01.2015 16:00.

[Spe15] Patrick Spenke. Konzeption und Umsetzung einer Balanced Scorecard zur Nutzenmessung anhand eines Lehr- und Lernportals im Rahmen des KOSMOS Projektes. Master's thesis, Universität Rostock, 2015. *Wird zum Zeitpunkt dieser Arbeit noch bearbeitet.*

[sql15] TRUNCATE TABLE (Transact-SQL). `https://msdn.microsoft.com/de-de/library/ms177570.aspx`, 2015. 24.01.2015 15.45.

[Str12] Bernhard Strehl. Einsatz von und probleme mit portalsoftware am beispiel einer universitären lehr-lernplattform. In *DeLFI*, pages 303–314. Citeseer, 2012.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[ur.15a] Fakultät für Informatik und Elektrotechnik der Uni Rostock - Wirtschaftsinformatik (B.Sc.). http://www.ief.uni-rostock.de/index.php?id=wirtschaftsinformatik-bachelor, 2015. 17.01.2015 17:10.

[ur.15b] LSF - Suche. https://lsf.uni-rostock.de/qisserver/rds?state=change&type=5&moduleParameter=veranstaltungSearch&nextdir=change&next=search.vm&subdir=veranstaltung&_form=display&function=search&clean=y&category=veranstaltung.search&navigationPosition=lectures%2Csearch&breadcrumb=searchLectures&topitem=lectures&subitem=search, 2015. 17.01.2015 17:10.

[weg15a] KOSMOS Wegtam Suchagent. http://wegtam-kosmos.informatik.uni-rostock.de/wegtam-kosmos/index, 2015. 09.01.201513:30.

[weg15b] Wegtam Search Agent - Suchmaschinen. http://wegtam-kosmos.informatik.uni-rostock.de/wegtam-kosmos/ose, 2015. 09.01.2015 13:55.

[weg15c] Wegtam Suchagent. https://wegtam.com/wegtam/index, 2015. 09.01.2015 13:00.

[weg15d] Wegtam Suchagent - Datenschutz. https://www.wegtam.com/wegtam/good-to-know, 2015. 09.01.2015 13:45.

[weg15e] wegtam.com - die Metasuchmaschine - Startups im Internet. http://www.startups-im-internet.de/internet-startups/wegtam-com-umfassend-und-anonym-das-web-durchsuchen/, 2015. 09.01.2015 13:35.

[Wei14] Tino Weigel. Joint-Editing Support for Academic Further Education - Cloud Linkage for MyKosmos. Master's thesis, Universität Rostock, August 2014.

[Wit59] Waldemar Wittmann. *Unternehmung und unvollkommene Information.* Springer, 1959.

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die Hausarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Sievershagen, den 02.02.2014