Hausarbeit

Persönliches Wissensmanagement mit dem elektronischen Zettelkasten

Stefan Schumacher

27. August 2021

Zusammenfassung

In dieser Hausarbeit beschreibe ich, wie ich mit einem selbst-entwickelten elektronischem Zettelkasten das notwendige Wissen für Projekte, Artikel und Bücher verwalte.

Ich zeige, wie ein Zettelkasten prinzipiell funktioniert, welche Vor- und Nachteile er im allgemeinen und in der elektronischen Version im speziellen hat.

Darüberhinaus zeige ich seine organisatorische und technische Einbindung in den Lebenslauf von mir verfasster Artikel und Bücher.

Entstanden in einem Seminar zum Wissensmanagement 2008, öffentliche Version.

Stand 2008.

Schlüsselwörter

Zettelkasten, Wissensmanagement, Lernkasten, Karteikarten, LaTeX, BibTeX

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

ABKÜRZUNGEN UND BEGRIFFE

Manche Menschen benutzen ihre Intelligenz zum Vereinfachen, manche zum Komplizieren.

(Erich Kästner)

BibTeX ..... Die Bibliographieverwaltung für LaTeX. Alle Werke werden in einer Textdatei in einer simplen Grammatik erfasst. Im Text werden lediglich der Zitierschlüssel und die Seitenzahl angegeben. Das Zitat im Text und das Literaturverzeichnis werden vollautomatisch generiert.

LaTeX ..... Ein Textsatzsystem. Als TeX von Donald E. Knuth entwickelt und von Laslie Lamport und vielen anderen erweitert. Der TeX-Quellcode wurde früher benutzt um Compiler zu testen.

Apache ..... *Der* Webserver. Ein Programm, um Webseiten in einem Netzwerk verfügbar zu machen.

Boolean ..... Binärer Datentyp, der nur 2 Werte annimmt (Ja/Nein), eine Art Schalter.

DBD/DBI ..... Database Driver/Interfacs, Perl-Schnittstelle für Datenbanken

Kollokation .... Linguistik: häufig benachbartes Auftreten von Wörtern auch: syntaktisches Bedeutungsfeld und lexikalische Solidaritäten

Log ..... Protokolldatei, Log-»Buch«

Multiversum ... Universum aus Universen.

Perl ..... *Die* Skriptsprache, insbesondere geeignet für Textverarbeitung

PostgreSQL .... Ein objektrelationales Datenbankmanagementsystem

Relation ..... Tabelle in einer Relationalen Datenbank

Repository ..... Eine Art Lager oder Depot, in dem Daten strukturiert abgelegt werden können. Im Bereich der Revisionskontrolle umfasst das Repository die eigentlichen Daten und deren Metadaten, also Zeitstempel, Benutzer, Status etc. pp.

Skript ..... Programm in einer *siehe* Skriptsprache hier ein Programm, das interaktiven Inhalt auf der Webseite erlaubt.

Skriptsprache .. Einfache Programiersprache für kleinere Aufgaben

SQL ..... Structured Query Language, Befehlssprache des Datenbanksystems PostgreSQL

Universum ..... Axiomatisch definierte Menge von Objekten.

1

EINFÜHRUNG

Nur wenige wissen, wie viel man wissen muss, um zu wissen, wie wenig man weiß.

(Werner Heisenberg)

1.1 VOM ZWECKE DIESER ARBEIT

Im Laufe meiner bisherigen Tätigkeit als Autor war es notwendig, eine Form des persönlichen Wissensmanagements einzusetzen.

Da ich stark projektorientiert arbeite, ist es notwendig, in den Projekten gemachte eigene Erfahrungen zu kondensieren. Darüberhinaus ist eine Form der Quellenverwaltung erforderlich, damit ich gegebenenfalls auf weiterführende Literatur zugreifen kann.

Beeinflusst durch mein bildungswissenschaftliches Studium, habe ich mein bisheriges System überarbeitet und am Zettelkasten-Prinzip ausgerichtet. Dabei habe ich das papierne-Karteikarten-Prinzip in Form einer Datenbank umgesetzt. Außerdem habe ich meine Erfahrungen im Schreiben von Artikeln und Büchern einfließen lassen und so versucht die technischen Arbeitsprozesse zu optimieren.

Der Autor einer wissenschaftlichen Arbeit geht in der Regel handwerklich so vor, dass er basierend auf einer Hypothese eine Grobgliederung der Arbeit erstellt. Anhand der Grobgliederung sucht er passende Literatur bzw. Quellen. Aus den Quellen exzerpiert er geeignete Zitate, die er in seine Arbeit unter Angabe der bibliographischen Daten einbaut.

Bei größeren Arbeiten bietet es sich an, einzelne Zitate in einem Zettelkasten zu sammeln. Dazu wird ein Zitat samt bibliographischer Daten auf einer Karteikarte erfasst. Während des Schreibprozesses kann das Zitat so bequem in die Arbeit aufgenommen werden. Darüberhinaus kann ein solcher Zettelkasten auch als »kondensiertes Gedächtnis« fungieren, wenn er über einen längeren Zeitraum gepflegt wird.

Diesen Prozess habe ich durch ein datenbankgestütztes System optimiert, welches ich in diesem Artikel beschreiben werde.

1.1.1 *Vom Schreiben*

Aufsätze, Artikel, Bücher und dergleichen lassen sich mit verschiedenen Werkzeugen erstellen. Füllfederhalter und Schreibpapier funktionieren einwandfrei, sind bei komplexeren Werken aber kompliziert zu gebrauchen. Wesentlich vereinfacht wird das Schreiben durch den Einsatz von Computern. Dafür existieren eine Vielzahl von Programmen, die mehr oder minder ihren Zweck erfüllen.

Zu den am wenigsten geeigneten – aber am verbreitetsten – Programmen gehören sogenannte Textverarbeitungsprogramme wie Microsoft Office, StarOffice, OpenOffice.org, Abiword, KOffice, TextMaker, InDesign, GEOS, PageMaker, WordPro, WordPerfect usw. Derartige Systeme sind in der Regel als sog. WYSIWYG[1]-Editoren konzipiert, d. h. der Autor arbeitet in einer graphischen Oberfläche, die das Schriftstück im zu druckenden Layout anzeigt. Dabei kommt es zwangsweise zu Abweichungen zwischen Druckstufe und Monitoransicht (z. B. durch die unterschiedliche Auflösung und Farbtiefe), die in der Regel aber ignoriert werden können.

Da das Office-Paket bzw. Word von Microsoft am verbreitetsten ist, werde ich mich hierauf als Vergleichsbasis einschränken.

Der Arbeitsablauf um eine wissenschaftliche Hausarbeit zu erstellen, weist unter MS Word erhebliche Probleme auf. Zum ersten ist der Autor während der Erstellung des Textes für das Layout zuständig. Er muss daher Überschriften, Bildunterschrift usw. während des Schreibens formatieren. Dies geschieht jedoch nicht durch Auszeichnung, sondern durch direktes formatieren. Der Autor markiert eine Überschrift eben nicht als Überschrift, sondern setzt diese manuell entsprechend, bspw. fett und zentriert. Dies hat erhebliche Verarbeitungsnachteile, da eine derartig erstellte Überschrift nur manuell umzuformatieren ist. Möchte der Autor alle Überschriften kursiv im linken Flattersatz darstellen, muss er jede einzelne Überschrift von Hand dahingehend abändern.

Neben der Satzauszeichnung fällt die Erstellung automatischer Verzeichnisse schwer. Zwar gibt es inzwischen Funktionen, um Abbildungs- und Tabellenverzeichnisse zu generieren aber selbstdefinierte Verzeichnisse, wie Glossare, Namens- oder Ortsverzeichnisse sind nicht automatisiert erstellbar. Sie müssen stattdessen vom Autor in Handarbeit generiert werden.

Einer der größten Schwachpunkte – neben der mangelnden Stabilität von Word und den gravierenden typographischen Mängeln – ist die Erstellung der Bibliographie. Word verfügt über keinerlei interne Mittel, um Literaturverzeichnisse und Zitate automatisch zu erstellen. Das heißt in der Praxis, das der Autor dies von Hand erledigen muss. Verwendet er dazu einen Zitierstil, der mehrmals die vollen bibliographischen Daten angibt, muss er diese auch mehrmals – also redundant – eintragen. Außerdem lässt sich der so erstellte »Zitierstil« nur von Hand wieder ändern.

All dies führt dazu, das es äußerst ineffizient ist, mit Word komplexere Artikel zu erstellen.

Eine funktionierende Alternative zu Office-Programmen ist LaTeX. Es handelt sich dabei nicht um einen WYSIWYG-Ansatz, sondern um WYSIWYM: *What you see is what you mean*. Der Text wird hierbei nicht direkt formatiert, sondern in einer Auszeichnungssprache ausgezeichnet und dann entsprechend vom Compiler formatiert. Der Autor markiert nicht mehr einen Text als fett und zentriert und somit als Überschrift, sondern er markiert den Text als Überschrift, zeichnet also seine *Semantik* aus. Der Compiler formatiert dann die Überschrift entsprechend der Einstellung zur Druckausgabe.

1 What you see is what you get

Dies hat mehrere entscheidende Vorteile. Zum einen wird Layout und Semantik getrennt. Eine Zeichenkette wird nicht mehr als Überschrift behandelt, weil sie in einem bestimmten Format formatiert ist, sondern weil sie als Überschrift ausgezeichnet wurde.

Damit ist es möglich, die vorliegenden Texte automatisiert weiterzuverarbeiten. Der Autor kann bspw. alle Überschriften aus einem Text heraussuchen lassen und an eine Datenbank verfüttern oder in eine Webseite integrieren lassen. Außerdem kann die Auszeichnung der Texte auch automatisiert durch Programme geschehen. So setzt z. B. die Fahrplanauskunft von www.NASA.de auf LaTeX. Möchte ein Benutzer einen Haltestellenfahrplan als PDF-Datei erstellen, werden die entsprechenden Daten aus der Datenbank herausgesucht, im LaTeX-Format ausgegeben und ins PDF-Format kompiliert. Somit ist es möglich, komplett automatisiert Tabellen oder ähnliches zu erstellen.

Diese Möglichkeit nutzt der Zettelkasten, in dem die Syntax (und daraus resultierend die Semantik) der Karteikarten in der Datenbankstruktur abgebildet werden. So gibt es z. B. Spalten für den Titel, die ID-Nummer, die Stichwörter usw.

Weitere Vorteile von LaTeX sind die Möglichkeiten, mit sehr geringem Aufwand eigene Verzeichnisse zu erstellen, wie bspw. die in der Geschichtswissenschaft üblichen Orts- und Personenverzeichnisse.

Eine der komfortabelsten Funktionen von LaTeX ist die Literaturverwaltung mit BibTeX. BibTeX folgt einem einfachen Prinzip, es hält alle Daten zum Literaturverzeichnis in einfachen Datenbanken vor und greift darauf bei Bedarf zurück. Das heißt, daß der Autor alle notwendigen Zitationsangaben zu einem Werk in einer einfachen Textdatei erfasst. Zitiert er aus dem Werk, gibt er einen speziellen LaTeX-Befehl an, der auf die Datenbank und das Werk zugreift. LaTeX bzw. BibTeX generieren dann automatisch die Zitation und das Literaturverzeichnis. Vorteile sind dabei, dass die Daten nur ein einziges mal erfasst werden müssen und Zitationsstile beliebig gewechselt werden können. Von APA bis DIN über Numerierung sind alle Varianten möglich und können durch einen einzigen Befehl gewechselt werden.

Der Autor muss daher nur noch seine Literaturdatenbank entsprechend pflegen, die Zitate abtippen und die Zitation im Artikel erstellen.

Da in den Sozialwissenschaften in der Regel mit sehr vielen Zitaten gearbeitet wird, möchte ich neben der Literaturdatenbank auch eine Zitatdatenbank benutzen. In der Zitatdatenbank sollen Zitate inkl. bibliographischer Angaben gespeichert werden, so das ich bei Bedarf das Zitat in einem Artikel einarbeiten kann. Zusätzlich soll die Datenbank durchsuchbar und verschlagwortbar sein.

Der elektronische Zettelkasten soll zusammengefasst folgende Kriterien erfüllen:

1. Offenes und Freies System, da die Daten über mehrere Jahrzehnte gepflegt werden sollen.
2. Anbindung an LaTeX und BibTeX, um Artikel samt Bibliographie zu erstellen.
3. Verschiedene Such- und Sortierfunktionen der Karteikarten.
4. Kategorisierung/Einordnung der Zitate.

5. Kategorisierung/Einordnung der Quellen.
6. Verwaltung von Hinweisen auf Quellen, die potenziell relevant sind. (Lesezeichenfunktion)
7. Verknüpfung der Zitate mit der Bibliographiedatenbank.
8. Kollaborationsmechanismen für mehrere Benutzer, um Gruppenarbeiten zu erleichtern.
9. Revisionskontrolle/Archivierung der Daten.
10. Unterstützung bei der Generierung eines semantischen Netzes.
11. Verwaltung der Zitate/Karten in einem Multiversum.
12. Client-Server-System.
13. Verteilter Zugriff über ein Webinterface.

Die Benutzung soll sich folgendermaßen gestalten:

1. Sammlung eines Zitats (Karteikarte anlegen)
2. Sammlung einer Quelle (Bibliographieeintrag anlegen)
3. Zitat mit Quelle verknüpfen
4. geeignete Zitate für einen Artikel heraussuchen
5. verknüpfte Quellen in die Bibliographie des Artikels aufnehmen.

Darüberhinaus soll der Zettelkasten während seiner Benutzung über mehrere Jahre hinweg gesammelte Zitate nicht nur archivieren, sondern mich auch beim Entdecken neuer Zusammenhänge und Verknüpfungen unterstützen. Aus dem Zettelkasten heraus sollen neues Wissen bzw. neue Artikel generiert werden können. Dazu werde ich im Laufe der Benutzung Hilfsfunktionen implementieren, die ich teilweise schon umgesetzt und in dieser Arbeit beschrieben habe.

2

DER ZETTELKASTEN

An jedem Punkt öffnet das Verstehen eine Welt.

(Wilhelm Dilthey)

2.1 KOMMUNIKATION MIT ZETTELKÄSTEN

Der Zettelkasten ist eine Form Wissen (bzw. in Artefakten kondensiertes Wissen) aufzubewahren. Sinn des Kastens ist es, als wichtig oder relevant erscheinende Textstellen, Zitate oder Aussagen zu speichern. Dazu wird auf genau einer Karteikarte genau eine Textstelle erfasst, mit der zitierfähigen Quellenangabe versehen und verschlagwortet. Die Schlagworte werden im Stichwortverzeichnis katalogisiert. Zu einem Stichwort werden die zugehörigen Karteikarten angegeben. Somit ist es möglich, über das Stichwortverzeichnis alle relevanten Karteikarten zu einem Thema zu finden.

Der Zettelkasten darf dabei nicht thematisch sortiert, sondern lediglich chronologisch numeriert werden. Da ein Zettelkasten durchaus mehrere Jahrzehnte in Gebrauch sein kann, ist nicht planbar welche Kategorien notwendig werden. Außerdem ist nicht immer klar, in welches Gebiet eine Karteikarte einsortiert werden soll. So passt z. B.die Frage »Was ist Lernen?« in die Gebiete Entwicklungspsychologie, Neuropsycholgie, Pädagogik und Philosophie.

Nachteilig ist die strikte Ordnung der Karten und die notwendige Verschlagwortung. Steht eine Karte am falschen Platz, kann sie nur mehr durch einen Zufall wiedergefunden werden. Ist eine Karte nicht verschlagwortet, steht sie nicht im Stichwortverzeichnis und kann daher nicht gefunden werden. Ebenso führt ein nicht vergebenes Schlagwort dazu, das die Kartei nicht im synaptischen Netz des Schlagwortes erscheint, da keine Querverbindung existiert.

Andere Verschlagwortungsmechanismen existieren, bspw. im Bibliothekswesen oder in der Verwaltungswissenschaft, sind für meine Belange aber nicht geeignet.

Benötige ich in einer Arbeit Quellen zu einem bestimmten Thema, kann ich anhand des Schlagwortverzeichnis relevante Karteikarten heraussuchen und ggf. zitieren.

2.1.1 *Design des Zettelkastens*

Alternativen

Ursprünglich habe ich meinen Zettelkasten in der klassischen Variante auf Papier geführt. Allerdings fiel mir schnell auf, das dies unbequem

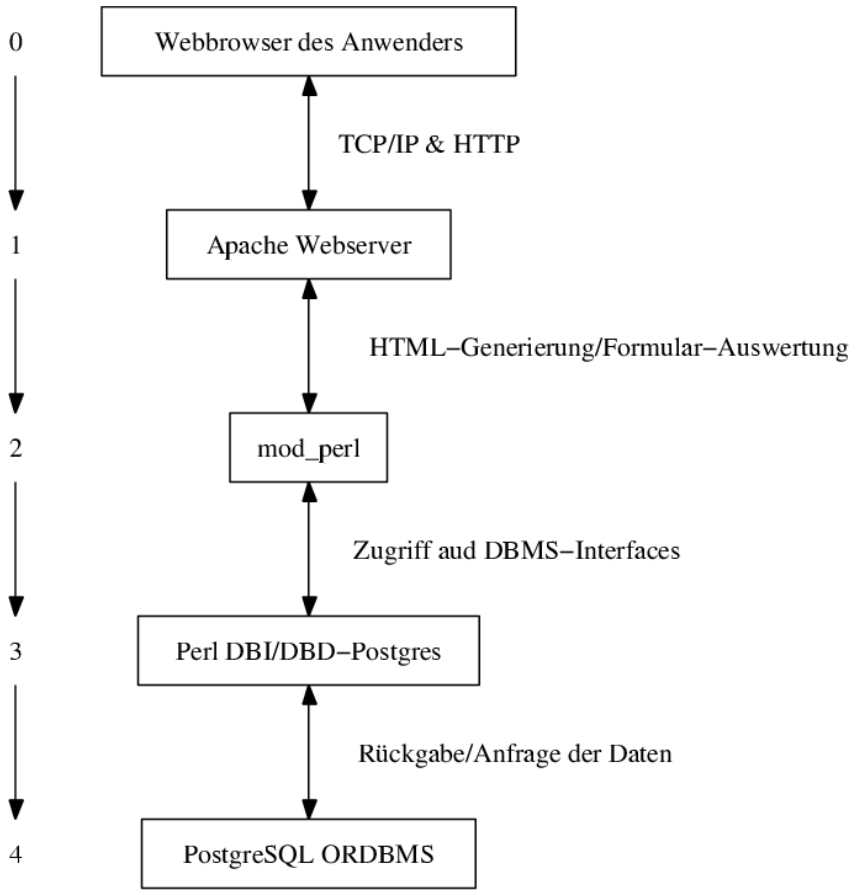

Abbildung 1: Hierarchie der Serverprogramme

ist. So musste ich einzelne Karteikarten wiederholt abtippen, um sie in mehreren Artikeln zu verwenden.

Daher wollte ich den Zettelkasten elektronisch führen. Bevor ich dazu eine eigene Entwicklung in Betracht gezogen habe, habe ich verschiedene Alternativen gesucht und ausprobiert. Allerdings konnte keine proprietäre Lösung meine Bedürfnisse befriedigen.

Es gibt kaum brauchbare Zettelkästen oder Wikis mit Anbindung an LaTeX und BibTeX, außerdem ist die Verwaltung der Daten und Metadaten oftmals aufwändig.

Da ich bereits einige Redaktionssysteme auf der Basis von PostgreSQL, Apache, Perl und LaTeX programmiert habe, ist ein grundlegendes Framework auf Basis der selbsterstellten Perl-Module bereits vorhanden. Daher war es nur notwendig, die eigentliche Kartenverwaltung und Anbindung an BibTeXzu entwerfen und zu implementieren.

Mittels Apache, Perl, DBD/DBI sowie CGI.pm stelle ich ein Webinterface her, über das der Zettelkasten gepflegt wird. Abbildung 1 zeigt die Hierarchie der Serverprogramme.

In einer PostgreSQL-Datenbank werden die Karteikarten, alte Versionen einer Karteikarte sowie die Bibliographie gepflegt. Alle Daten liegen somit innerhalb der PostgreSQL-Datenbank vor. Lediglich die Bibliographiedaten liegen in der PostgreSQL-Datenbank und der BibTeX-Datenbank vor, sind also redundant. Da BibTeX (noch) nicht direkt auf PostgreSQL zugreifen kann, ist dies notwendig.

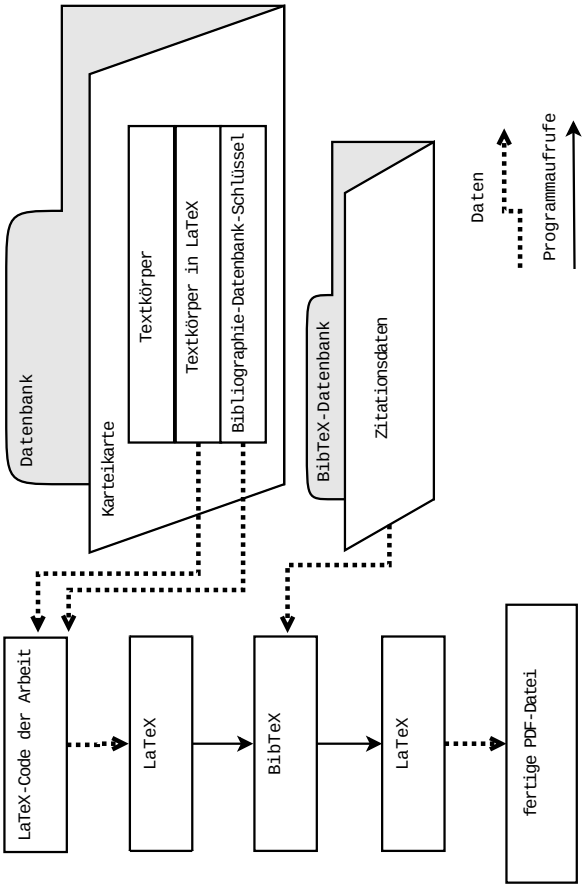

Abbildung 2: Logische Struktur des Zettelkastens

Die Datenstruktur[1] des Zettelkastens ist vergleichsweise einfach, sie besteht aus folgenden Relationen:

BEARBEITER Enhält Daten zum Bearbeiter der Karte, wie User-Name, Gruppen-Name und Login. Der Zettelkasten ist für Mehrbenutzerbetrieb ausgelegt, die entsprechenden Benutzerdaten liegen in dieser Tabelle.

PRAED In dieser Tabelle können Stichwörter und bilbiographische Angaben voreingestellt werden. In jeder neuen Karte werden diese Daten automatisch übernommen.

BIBLIOGRAPHIE enthält die bibliographischen Daten, wie Zitierschlüssel, Autor, Titel usw. Tabelle 2 zeigt die Datenstruktur der Relation bibliographie.

ZETTELKASTEN enthält die eigentlichen Karteikarten. Tabelle 1 zeigt die Datenstruktur der Relation zettelkasten.

ZETTELKASTEN.BAK archiviert bei einem Update die alte Version der Karte. Sie ist funktional äquivalent zur zettelkasten-Relation

1 Vereinfacht gesagt besteht eine relationale Datenbank aus mehreren Tabellen, die miteinander verknüpft werden. Ein einfaches Beispiel sind die alten Prüfungslisten. In einer Tabelle stehen die Daten der Studenten inkl. Matrikelnummer, in einer zweiten Tabelle die Matrikelnummer samt Note in der Klausur. Über die Matrikelnummer können die Daten in Beziehung (Relation) gesetzt und aufgelöst werden. Da die Matrikelnummern der Studenten im Netz veröffentlicht wurden, werden nun Prüfungsnummern verwendet. Das Prinzip ist aber das selbe.

Datum	Zweck
datum	Zeitstempel der Karte
ident	ID-Nummer, Primärschlüssel
titel	Titel der Karte
textkoerper	Textkörper bzw. Inhalt der Karte
bibkey	Schlüssel des zitierten Werkes
seite	Seitenangabe des Zitats
bibkeynach	Schlüssel des Werkes, nach dem zitiert wird
seite_zit_nach	Seitenangabe des Zitates nach
bib_ausserhalb	Werk außerhalb der Bibliographie
random_id	automatisch generierte Zufallszahl
stichwoerter	Die Stichwörter der Karte
version	Versionsnummer der Karte
cvslog	Logeintrag für neuere Versionen

Tabelle 1: Die Relation zettelkasten

Datum	Zweck
key	Zitierschlüssel des Werkes, Primärschlüssel
autor	Autor
titel	Titel
type	Typ (Buch, Artikel, Tagungsband etc.)
ident	laufende Nummer
jahr	Erscheinungsjahr

Tabelle 2: Die Relation bibliographie

und enthält lediglich weitere Felder für einen Zeitstempel, einen Logeintrag und die Versionsnummer der alten Karte.

Der Benutzer des Zettelkasten kommuniziert über Webformulare mit ihm, die ich im folgenden vorstellen werde:

Eine Karteikarte anlegen oder bearbeiten

Um eine Karteikarte anzulegen, öffne ich das entsprechende Formular und trage den Titel, den Textkörper (das eigentliche Zitat), die Stichwörter und die Quellenangabe ein. Ist eines dieser Felder leer, wird die Karteikarte nicht angelegt.

Für die Bibliographieschlüssel stehen 2 Pull-Down-Menüs zur Verfügung, in der Ich die entsprechenden Quellen auswählen kann. Außerdem kann er in einem Textfeld eine neue Quelle angeben, bspw. eine URL.

Da eine Karteikarte fehlerhaft sein kann, ist es notwendig, eine Korrekturmöglichkeit einzubinden. Dazu wird das selbe Formular wie zum Erstellen der Karte benutzt (karteikarte-bearbeiten.pl). Zum Bearbeiten einer Karte übernimmt es deren ID-Nummer, liest die Daten aus der Datenbank aus und befüllt damit die Textfelder des Bearbeiten-Formulars. Zusätzlich wird ein Textfeld für einen Logeintrag eingebunden, in dem ein kurzer Kommentar zum Grund der Änderung eingetragen werden kann, wie bspw. Tippfehler, falsche

Seitenzahl o.ä. Somit kann ich eine Karte beliebig korrigieren und wieder abspeichern.

Bevor die Karteikarte in der Datenbank aktualisiert wird, wird der alte Zustand als neuer Datensatz in die Relation zettelkasten_bak kopiert. In der aktualisierten Karte wird der Versionszähler um 1 erhöht. Gibt es also eine Karteikarte mit der Versionsnummer 4 im Zettelkasten, dann sind die Versionen 1, 2 und 3 im Archiv gespeichert und können bei Bedarf angezeigt werden.

Da ich in der Regel aus einem Buch mehrere Karteikarten erstelle und dabei nahezu immer die selben Stichwörter verwende, habe ich eine Zwischenspeicherfunktion erstellt. Dazu kann ich im Formular »Prädisposition definieren« Stichwörter und Bibliographieschlüssel eingeben. Diese werden anschließend in der praed-Relation zwischengespeichert und bei jeder neuen Karte automatisch übernommen, bis sie mit dem Formular »Prädisposition löschen« entfernt werden. Dadurch spare ich Schreibarbeit ein.

Karteikarten anzeigen bzw. durchsuchen

Die wichtigste Funktion des Zettelkastens ist es, Karteikarten anzuzeigen bzw. zu suchen.

Auf der Startseite des Zettelkastens werden daher die Nummer und der Titel aller Karteikarten angezeigt. Durch einen Klick auf den Titel wird der gesamte Karteninhalt angezeigt. Neben dem Titel wird noch der Direktlink zum Bearbeiten sowie ein Link zum LaTeX-Code angezeigt.

Auf der Anzeige-Seite werden folgende Daten angezeigt:

- die Ident-Nummer der Karte,
- die Versionsnummer,
- der Titel,
- der Zeitstempel,
- der Textkörper,
- die Quelle (Autor, Titel, Jahr, Seiten),
- die Stichwörter
- Links auf die Archivversionen, das Skript zum Bearbeiten und den LaTeX-Code

Die Stichwörter werden direkt als Link eingebunden, d. h. klickt man auf ein Stichwort, bekommt man eine Liste aller Karten, die ebenfalls so verstichwortet wurden. Durch diese Funktion kann ich ein semantisches Netz zwischen den Karten spannen. Aber dazu mehr im 3 Kapitel.

Das LaTeX-Code-Skript zeigt den Textkörper und die Quellenangabe der Karte als LaTeX-Code an, so das ich mit der Maus und Copy-&-Paste das Zitat in einem Artikel übernehmen kann.

Um den Zettelkasten zu durchsuchen, habe ich 2 Suchfunktionen eingebaut: »Stichwörter durchsuchen« und »Karteikarten durchsuchen«. Beide Funktionen übernehmen über ein Textfeld eine zu

suchende Zeichenkette und durchsuchen damit entweder nur die Stichwörter oder die gesamte Karte. Dadurch ist es, anders als im klassischen Zettelkasten, möglich, jede Karteikarte immer wieder zu durchsuchen und ggf. neu zu indizieren. Trotzdem ist es notwendig, neue Karteikarten sorgfältig und konsistent zu verstichworten und bspw. nicht zwischen dem deutschen »soziale Kompetenzen« und dem Buzzword »soft skills« hin und her zu springen.

Die Suche nach den Zeichenketten übernimmt PostgreSQL über den ~*-Operator, der Reguläre Ausdrücke zur Verfügung stellt. Damit ist es möglich, unscharfe Suchen[2] mittels . und * durchzuführen.

Eine weitere Möglichkeit Karteikarten anzuzeigen, ist die »Glückskeks-Funktion«. Diese wählt zufällig 5 Karten aus und zeigt sie auf einer Seite an. Dadurch kommte der Pseudozufall ins Spiel und ermöglicht evolutionäre Sprünge im semantischen Netz der Karten durch zufällig mögliche Verknüpfungen.

Die Funktion »Alle Stichwörter anzeigen« erstellt eine Liste aller Stichwörter, die im Zettelkasten vorkommen. Durch klick auf das Stichwort, werden alle Karteikarten angezigt, die mit dem Stichwort markiert sind.

Die Druckversion

Da ich die Karteikarten nicht nur elektronisch, sondern auch als Papierversion haben möchte, benötige ich eine Druckversion der Karteikarten.

Hierzu sind Din-A6-Karteikarten geeignet, die mit LaTeX aufbereitet werden.

Ein Perlskript liest den Datenbankinhalt aus und wandelt ihn in LaTeX-Code um. Die Kopf- und Fußdaten (Dokumententyp, Schriftgröße, etc. pp.) werden aus 2 Textdateien ausgelesen und an den Anfang bzw. das Ende der Datei gesetzt.

Die Quellenangaben werden über den Zitierstil `verbose` komplett auf jeder Karte ausgegeben. In früheren Versionen habe ich den klassischen Autor-Jahr-Zitiertstil – wie in dieser Hausarbeit – verwendet, also bspw. *Feyrer, Schumacher und Weinem (2007, S. 319)* auf der Karteikarte und am Ende der Karten einige Karten, auf denen die gesamten bibliographischen Daten angegeben waren.

Das hatte aber mehrere Nachteile. Zum einen musste ich immer alle Karten drucken, wenn neue hinzugekommen sind, da sich die Bibliographie am Ende der Karten meist geändert hat. Zudem war eine Karte für sich genommen nicht zitierfähig, da die genauen bibliographischen Daten fehlten. Diese Probleme lassen sich mit dem Vollzitat auf jeder Karte umgehen.

Mittels `makeindex` erstellt LaTeX automatisch ein Stichwortverzeichnis, in dem die Stichwörter der Karteikarten und die entsprechende Seitenzahl erfasst sind, Abb. 3 zeigt das Stichwortverzeichnis.

2 Dabei werden mit einem Punkt ein einzelnes Zeichen und mit dem Asterisk beliebig viele Zeichen ausgelassen. `Kompetenz.*` träfe also bspw. Kompetenz, Kompetenzen, Kompetenzentwicklung, Kompetenzdiagnostik usw.

Abbildung 3: Das Stichwortverzeichnis mit den Verweisen auf die Karten

94^2 Z-Transformation **2008-08-29**

$$z_i = \frac{x_i - \bar{x}}{S_x} \tag{5}$$

$$\bar{z} = \sum_{i=1}^{n} \frac{x_i - \bar{x}}{S_x} = 0 \tag{6}$$

$$S_z = 1 \tag{7}$$

$$F(x) = \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot \int_{-\infty}^{x} \mathrm{e}^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2} \mathrm{d}t \tag{8}$$

$$= \frac{1}{\sigma \cdot \sqrt{2\pi}} \cdot \int_{\frac{-\infty-\mu}{\sigma}}^{\frac{x-\mu}{\sigma}} \mathrm{e}^{-\frac{1}{2}u^2} \mathrm{d}u \cdot \sigma \tag{9}$$

$$= \frac{1}{\sqrt{2\pi}} \cdot \int_{-\infty}^{\frac{x-\mu}{\sigma}} \mathrm{e}^{-\frac{1}{2}u^2} \mathrm{d}u \tag{10}$$

$$= \Phi\left(\frac{x-\mu}{\sigma}\right) \tag{11}$$

Quelle: Vorlesung Quantitative Forschungsmethoden SSo8 (Lühnenschloß/Edelmann-Nusser)
Siehe auch: Statistik; Forschungsmethoden, Quantitative; Z-Transformation; ;

© Stefan Schumacher www.bildungswissenschaft.info

Abbildung 4: Der exzellente Mathematik-Satz unter LaTeX gezeigt anhand der Z-Transformation

2.1.2 *Lernkarten*

Der Zettelkasten lässt sich ebenfalls als Lernkasten nutzen.

Eine einfache Variante ist die Lernkartei mit 5 Fächern, die sich hervorragend zum Auswendiglernen von Vokabeln, Definitionen oder psychologischen Theorien eignet.

Dazu wird ein Karteikasten mit 5 Fächern benutzt. Auf den Karten stehen auf der Vorderseite die Vokabeln bzw. Fragen, auf der Rückseite die passende Antwort.

In den ersten Kasten kommen alle neuen Definitionen, die noch zu lernen sind. Diese Karten werden jeden Tag herausgenommen und bearbeitet. Dazu liest man die Frage und überlegt die passende Antwort. Nachdem man die (vermeintlich richtige) Antwort gegeben hat, dreht man die Karte um und vergleicht die gegebene Antwort mit der auf der Karte stehenden.

War die gegebene Antwort korrekt, wird die Karte ins 2. Fach gesteckt, War die Antwort falsch, kommt sie zurück ins 1. Fach.

Täglich wird das 1. Fach durchgearbeitet, das 2. kommt erst an die Reihe, wenn es fast voll ist. Dies geht der Reihe nach durch, bis man am (theoretischen) Ende alle Karten in das 5. Fach sortiert hat und das Themengebiet dadurch gefestigt gelernt wurde.

Dies strukturiert neue Lerninhalte nach Gedächtnisleistung und optimiert vor allem die Leistung bei auswendig zu lernenden Inhalten (vgl. Mietzel, 2001; Niegemann et al., 2003; Reischmann, 2002; Spitzer, 2006).

Da insbesondere in den Grundlagenvorlesungen der Psychologie viele Theorien gelernt werden müssen, eignet sich eine Kopplung zwischen Zettelkasten und Lernkartei. Lernkarten kann ich so direkt aus dem Zettelkasten generieren.

Daher habe ich den Zettelkasten um eine Lernkarteifunktion erweitert. Da es nun nicht mehr Karteikarten, sondern auch Lernkarten gibt, wird aus dem Universum »Zettelkasten« ein Multiversum.

Dies bedeutet, das es nun Datensätze für Lernkarten und Karteikarten gibt. Da sich die Lernkarten von den Karteikarten unterscheiden, habe ich eine 2. Relation `lernkasten` eingefügt. Auf einer Lernkarte findet sich das entsprechende Fach und Semester (z. B. Sozialpsychologie 1, Entwicklungspsychologie 2 usw.), eine laufende Nummer und Datum sowie die Frage. Auf der Rückseite wird die entsprechende Antwort angegeben. Die Abbildungen 5 und **??** zeigen die Lernkarte 7.

Die Datenstruktur entspricht etwas abgespeckten Karteikarten, daher sind auch die Skripte zum hinzufügen/ändern der Lernkarten nahezu die selben.

Die Lernkarten lassen sich sowohl ausgedruckt benutzen, als auch in einer Online-Version. Für die Onlineversion habe ich die Variable `lernfach` hinzugefügt, in der das entsprechende Lernfach (1-5) abgespeichert wird. Dadurch kann ich die »Holzvariante« in der Datenbank simulieren. Über ein Webinterface werden der chronologischen Reihe nach die Lernkarten des ersten Fachs angezeigt. Erst erscheint die Frage und danach in einem weiteren Skript die Antwort. Ist die Antwort richtig, kann ich die Lernfachnummer erhöhen, die Karte also weitersortieren.

Abbildung 5: Lernkarte 7 zeigt die Vorderseite einer Lernkarte im A6-Format

2.1.3 *Eine Bibliographie pflegen*

Wissenschaftliche Artikel erfordern Quellenangaben. Diese beinhalten mindestens den oder die Autoren, den Titel des Werkes, den Titel des Buches oder Journals und das Veröffentlichungsjahr. Weitere Angaben, wie URL, Verlag und Verlagsort, Seitenzahlen, Auflage usw. kommen meist noch hinzu.

Daher ist es sinnvoll, derartige Quellenangaben zentral in einer Datenbank zu erfassen. Für LaTeX übernimmt BibTeX diese Aufgabe. Dazu erfasse ich die bibliographischen Angaben in einer Textdatei in Form einer einfachen Grammatik.

Diese Textdatei wird im Dokument eingelesen und aufbereitet. Im Text kann mittels "cite-Befehl eine Zitationsangabe erstellt werden. BibTeX erzeugt dann automatisch die Zitationen im laufenden Text und das Literaturverzeichnis.

So wird aus dem Befehl "textcite[][S.",319]–osjb07" im Text automatisch »Feyrer, Schumacher und Weinem (2007, S. 319)«, außerdem erzeugt BibTeX automatisch den zugehörigen Bibliographieeintrag:

> Feyrer, Hubert, Stefan Schumacher und Mark Weinem (2007). »NetBSD – Das portabelste Betriebssystem der Welt«. In: Open Source Jahrbuch 2007. Herausgegeben von Bernd Lutterbeck, Matthias Bärwolff und Robert A. Gehring. Berlin: Lehmanns Media, Seiten 315 – 326. isbn: 978-3-86541-191-4. url: http://www. opensourcejahrbuch.de/download/jb2007/OpenSourceJahrbuch2007online.pdf (besucht am 2007. 03. 17).

Seit einiger Zeit verwende ich BibLaTeX, eine Art Style für BibTeX, der weitere Konfigurationen erlaubt. Besonders nützlich sind hierbei die erweiterten Felder abstract für eine Kurzzusammenfassung des Werkes und keyword, mit dem ein Eintrag verschlagwortet werden kann.

In normalen Bibliographien werden diese Felder nicht ausgegeben. Ich verwende sie aber in speziellen Stilen, bspw. um eine kommentierte Bibliographie für den Zettelkasten zu erstellen oder eine Bibliographie zu einem bestimmten Themenkreis.

Mit dem Schlüsselwort *eigen* oder *reader* markiere ich bspw. alle Werke die ich selbst besitze oder zumindest als Seminarreader vorliegen habe. Somit ist es einfach möglich thematisch sortierte Bibliographien zu erstellen.

Zusammen mit dem abstract-Feld lassen sich so auch komfortabel Werke erfassen, die ich noch nicht gelesen oder zumindest exzerpiert habe, wie bspw. Zeitschriftenartikel oder Werke, die in anderen Werken zitiert werden.

Somit lässt sich ein Netz aus Quellen aufbauen, die für zukünftige Artikel von Bedeutung sein können.

Abb. 6 zeigt drei BibLaTeX-Einträge, Abb. 7 das Ergebnis im reading-Stil.

```
@inproceedings–ffg06,
 author = –Stefan Schumacher″,
 title = –Methoden zur Datensicherung --
 Strategien und Techniken für NetBSD″,
 year = –2006″,
 publisher = –GUUG e.V.″,
 booktitle="Proceedings des GUUG Frühjahrsfachgespräches 2006",
 address="Köln",
 howpublished="Vertrieben durch Lehmanns Fachbuchhandlung",
 pages="79-98",
 isbn="3-86541-145-2",
 keyword="Schumacher, eigen, Backup, Bacula, Datensicherung,
  FFG, GUUG",
″

@incollection–osjb07,
 author = "Hubert Feyrer and Stefan Schumacher and Mark Weinem",
 title = "NetBSD -- Das portabelste Betriebssystem der Welt",
 pages = "315 -- 326",
 booktitle = "Open Source Jahrbuch 2007",
 editor="Bernd Lutterbeck and Matthias Bärwolff
  and Robert A. Gehring",
 isbn="978-3-86541-191-4",
 address="Berlin",
 publisher="Lehmanns Media",
 year="2007",
 url="http://www.opensourcejahrbuch.de/download/jb2007/
  OpenSourceJahrbuch2007˙online.pdf",
 urldate="17.03.2007"
 keyword="Schumacher,eigen, NetBSD, Open Source",
″


@article–datenschleuder,
 author = –Stefan Schumacher″,
 title = –Systeme mit Systrace härten″,
 year = –2007″,
 issue="91",
 howpublished= –Mitgliederzeitschrift des Chaos Computer Clubs″,
 journal="Die Datenschleuder",
 address="Hamburg",
 pages="40-48",
 issn="0930-1054",
 url="http://ds.ccc.de/pdfs/ds091.pdf",
 urldate="03.01.2008",
 keyword="Schumacher, NetBSD, Systrace, CCC",
 abstract="Dieser Artikel beschreibt die Funktionsweise von
  Systrace, Aufbau und Erzeugung einer Richtlinie sowie den
  praktischen Einsatz anhand von zwei Beispielen auf NetBSD.",
″

```

Abbildung 6: Beispiel für BibLaTeX-Einträge

Literaturverzeichnis

Feyrer u. a.: NetBSD – Das portabelste Betriebssystem der Welt **osjb07**

Hubert Feyrer, Stefan Schumacher und Mark Weinem. „NetBSD – Das portabelste Betriebssystem der Welt“. In: *Open Source Jahrbuch 2007*. Herausgegeben von Bernd Lutterbeck, Matthias Bärwolff und Robert A. Gehring. Berlin: Lehmanns Media, 2007, Seiten 315 –326. ISBN: 978-3-86541-191-4. URL: http://www.opensourcejahrbuch.de/download/jb2007/OpenSourceJahrbuch2007_online.pdf (besucht am 2007.03.17).

Schumacher: Methoden zur Datensicherung – Strategien und Techniken für NetBSD **ffg06**

Stefan Schumacher. „Methoden zur Datensicherung – Strategien und Techniken für NetBSD“. In: *Proceedings des GUUG Frühjahrsfachgespräches 2006*. Köln: GUUG e.V., 2006, Seiten 79–98. ISBN: 3-86541-145-2.

Schumacher: Systeme mit Systrace härten **datenschleuder**

Stefan Schumacher. „Systeme mit Systrace härten“. In: *Die Datenschleuder* (91 2007), Seiten 40–48. ISSN: 0930-1054. URL: http://ds.ccc.de/pdfs/ds091.pdf (besucht am 2008.01.03).

Zusammenfassung: Dieser Artikel beschreibt die Funktionsweise von Systrace, Aufbau und Erzeugung einer Richtlinie sowie den praktischen Einsatz anhand von zwei Beispielen auf NetBSD.

Abbildung 7: Beispiel für BibLaTeX-Einträge im Literaturverzeichnis

2.1.4 *Die Gedankenmüllhalde*

Oftmals ist es für mich sinnvoll, eigene Gedanken, Assoziationen, Ideen und Geistesblitze zu kondensieren. Dies tue ich normalerweise in einem Notizbuch. Allerdings ist dieses Buch chronologisch sortiert, es sei denn ich verwende direkt Karteikarten. Daher ist es oftmals kompliziert, die Beziehungen zwischen den einzelnen Notizen herzustellen.

Da die Notizen früher oder später sowieso mit meinem Zettelkasten verheiratet werden, kann ich sie auch direkt im Zettelkasten-Universum führen.

Eine bequeme Möglichkeit ist es daher, den Zettelkasten als Gedankenmüllhalde zu benutzen. Dabei erfasse ich einzelne Gedanken auf einer Karteikarte und verstichworte sie entsprechend. Quellenangaben sind nicht notwendig, können also weggelassen werden, es sei denn, ich möchte direkt auf ein Werk verweisen.

Die Gedanken werden ebenfalls auf einer Karteikarte erfasst, die über den Schalter zitierfaehig als nicht-zitierfähig bzw. als »Gedanke« markiert wird.

Durch diese konsequente Erfassung ekklektizistisch generierter Gedanken entstehen einzigartige semantische Netzwerke. Die Netzwerke stellen so neue Verknüpfungen her, die sonst nie entstanden wären.

Im Rahmen meiner Gedankenmüllhalde zum Thema »Was ist Sicherheit?« verknüpften sich im Laufe der Zeit unter anderem Ulrich Becks »Weltrisikogesellschaft« mit Friedrich Spees »Cautio Criminalis«, Paul Watzlawicks 2 Realitäten mit Max Webers protestantischer Ethik und Neuropsychologische Erkenntnisse über Oxytocin und den Nucleus Paraventricularis mit dem Streit um das katholische bzw. protestantische Heilsversprechen.

2.1.5 *Artikelentwürfe*

Des öfteren entstehen aus Ideen der Gedankenmüllhalde weitere Ideen bzw. Grundlagen für neue Artikel, so z. B. momentan der Entwurf, mittels Jean Piagets genetischer Epistemologie, die Unmöglichkeit künstlicher »Intelligenzen« nachzuweisen. Diese Idee verknüpft also entwicklungspsychologische Theorien mit der Informatik.

Inzwischen sammle ich die Ideen zu einem Artikel in einem Artikelentwurf. Er enthält in loser Form Stichpunkte, Ideen und Skizzen. Außerdem Verweise auf Karteikarten und Quellen in der Bibliographie.

Somit habe ich in der Planungsphase die Möglichkeit, alle relevanten Informationen zu sammeln und diese bei Projektbeginn verfügbar. Die fehlende Strukturiertheit des Zettelkastens wird hier wieder zu einem sinnvollen Werkzeug der Datenorganisation.

So kann ich bspw. Stichwörter und Literaturangaben sammeln und erweitern und zu Beginn des Artikels eine entsprechende Literaturliste erstellen, die sich abarbeiten lässt.

2.1.6 *Projektmanagement/Wissensmanagement*

Der Zettelkasten lässt sich so auch als Hilfsmittel im Wissens- bzw. Projektmanagement einsetzen. So erfasse ich Notizen zu einzelnen Projekten auf Karteikarten und ordne sie Projekten bzw. Themen/ Kategorien zu. Dies ermöglicht mir, »Wissen« zu archivieren und bei Bedarf darauf zuzugreifen.

Ursprünglich habe ich den Zettelkasten in leicht abgewandelter Form als Projektmanagementsystem für die Organisation des Magdeburger Open-Source-Tages verwendet. Da es mit dem Zettelkasten aber nicht möglich ist, z.B. Arbeitspakete zu erstellen und zu verteilen oder Termine zu planen, habe ich das Managementsystem auf TaskJuggler bzw. Redmine umgestellt. Letzteres ist ebenfalls ein webbasiertes System, so das die gesamte Orga-Truppe problemlos Aufgaben und Termine koordinieren kann.

2.2 Revisionskontrolle mit CVS

Da sich die Arbeit an einem Buch oder Artikel über mehrere Monate oder Jahre hinziehen kann, ist eine Revisionskontrolle der Dateien sinnvoll. Eine Revisionskontrolle ist ein Verwaltungssystem für Daten und Metadaten. Es dient daher folgenden Aufgaben:

- Vorhalten aller Versionen einer Datei
- Protokolle in Log-Dateien ablegen
- einzelne Dateien für einen Autor sperren
- verschiedene Benutzer verwalten und deren Aktivitäten protokollieren
- ein zentrales Repository zur Verfügung stellen

Somit lege ich in der Revisionskontrolle alle Metadaten über den Lebenszyklus der Dateien ab. Außerdem vereinfacht CVS die Arbeit mit mehreren Autoren. Da das CVS-Repository die Daten zentral vorhält, können darüber die Arbeiten der Autoren synchronisiert werden. Jeder Autor aktualisiert zu Beginn seiner Arbeit seine lokalen Dateien vom Server und lädt sie nach der Arbeit wieder auf den Server hoch. Dort stehen sie dann für andere Autoren bereit und können von diesen wieder runtergeladen werden. Dieses Hub-Prinzip vereinfacht die Koordinierung mehrerer Autoren ungemein.

Besonders interessant ist das sogenannte Branching, dabei werden verschiedene Entwicklungszweige erstellt. Diese Entwicklungszweige können unabhängig voneinander weitergepflegt werden. Das Branching entstammt ursprünglich der Softwareentwicklung, wo verschiedene Softwareversionen parallel nebeneinander gepflegt wurden. Es zeigt seine Stärken bei umfangreichen Projekten, die aus mehreren Dateien bestehen. So sind bspw. aus meinem NetBSD-Sicherheitshandbuch seit 2005 diverse Artikel und Vorträge hervorgegangen, die allesamt auf dem Originaltext des Handbuchs basieren. Da ich die Dateien allesamt in einem CVS-Repository vorhalte, ist es sehr einfach, Artikel für Zeitschriften abzuspalten, zu überarbeiten und wiedereinzufügen.

Abb. 8 zeigt eine Log-Ausgabe und somit die Entwicklungsgeschichte einer Hausarbeit. Die Zeile ABGABE: 1.12.0.2 zeigt, das ich Version 1.12.0.02 vom 29.04.2008 abgegeben habe, Version 1.13 vom 04.02.2009 wurde mit EingereichtZurUpTimes markiert und von mir zur Veröffentlichung in der Zeitschrift UpTimes eingereicht. Somit existieren jetzt zwei Versionen nebeneinander, einmal die klassische Hausarbeitsvariante und die überarbeitete Version, die in der UpTimes erscheinen wird. Beide Versionen lassen sich parallel weiterpflegen, so kann ich in beiden Zweigen die jeweiligen Korrekturen der Redaktion bzw. Dozentin einarbeiten.

Abb. **??** zeigt die Zweige eines größeren Projektes. Die Zweige entstanden im Laufe der Entwicklung, CLT07 ist bspw. eine Version, die ich während eines Vortrag auf den Chemnitzer Linux-Tagen 2007 eingesetzt habe.

Zwar verwende ich CVS nicht direkt für den Zettelkasten, aber einige Funktionen (Versionierung, Log, Autorenmarkierung) habe ich in der Datenbank implementiert. Außerdem halte ich *jede* Hausarbeiten oder Artikel in einem CVS-Repository vor.

Ähnlich der Logfunktion der Datenbank ermöglicht die Logfunktion des CVS-Repositories, Metadaten auszuwerten. Die gesamte Entwicklungsgeschichte wird (zumindest in Teilen) kondensiert und kann so ausgewertet werden.

```
1
2  RCS file: /home/daten/cvs/SoKoInf/Soziale˙Kompetenzen˙fuer˙Informatiker.tex,v
3  Working file: Soziale˙Kompetenzen˙fuer˙Informatiker.tex
4  head: 1.13
5  branch:
6  locks: strict
7  access list:
8  symbolic names:
9      EingereichtZurUpTimes: 1.13.0.2
10     ABGABE: 1.12.0.2
11     current: 1.1.1.1
12     stefan: 1.1.1
13 keyword substitution: kv
14 total revisions: 14;    selected revisions: 14
15 description:
16 ----------------------------
17 revision 1.13
18 date: 2009/02/04 22:22:06;  author: stefan;
19 state: Exp;  lines: +104 -61
20 Aufbereitung zur Veröffentlichung
21 ----------------------------
22 revision 1.12
23 date: 2008/04/29 06:59:22;  author: stefan;
24 state: Exp;  lines: +114 -38
25 Endversion vor Abgabe
26 ----------------------------
27 revision 1.11
28 date: 2008/04/27 18:51:25;  author: stefan;
29 state: Exp;  lines: +213 -211
30 Erster Korrekturlauf
31 ----------------------------
32 revision 1.10
33 date: 2008/04/21 13:05:54;  author: stefan;
34 state: Exp;  lines: +49 -105
35 Entstehung von k.Verhalten, kleine Aufräumarbeiten
36 ----------------------------
37 revision 1.9
38 date: 2008/04/21 09:33:01;  author: stefan;
39 state: Exp;  lines: +163 -32
40 Verfahren und Verhaltenssteuerung
41 ----------------------------
42
43 [...]
44
45 ----------------------------
46 revision 1.1.1.1
47 date: 2008/01/09 22:50:09;  author: stefan;  state: Exp;  lines: +0 -0
48 Beginn der HA
49 =============================================================================
50
51
```

Abbildung 8: Das CVS-Log meiner Hausarbeit über Soziale Kompetenzen für Informatiker

3

DAS SEMANTISCHE NETZ

Jede Notiz ist nur ein Element, das seine Qualität erst aus dem Netz der Verweisungen und Rückverweisungen erhält.

(Niklas Luhmann)

Bisher habe ich nur die technischen Details des Zettelkastens beschrieben, ohne seine eigentliche Stärke zu zeigen. Er ist nicht nur ein einfaches Werkzeug, um Zitate und Definitionen wie ein Lexikon abzuspeichern, sondern er ermöglicht es, neues Wissen im semantischen Netz zu generieren.

Das semantische Netz besteht dabei aus den Verknüpfungen zwischen den Karten (den Verweisen bzw. Synapsen) und der *Bedeutung* bzw. der *Semantik* des Verweises. Dies führt dazu, das jedes Element im Netz, sei es Knoten (Karteikarte) oder Kante (Verweis) Wissen erzeugt und so neues Wissen generieren kann.

Um ein solches Netz aufzuspannen, verschlagworte ich die Karteikarten. Über die Schlagwörter lassen sich die korrespondierenden Karteikarten finden. Außerdem ermöglichen die von mir so genannten Kollokationen Sprünge aus einem Schlagwortnetz heraus. Eine Kollokation entsteht, wenn zwei unterschiedliche Schlagwörter auf einer Karteikarte auftauchen. Sie sind dann über diese Karte verbunden und spannen einen Pfad auf, der den Raum des ursprünglichen Schlagwortes verlässt.

Abb. **??** zeigt alle Karteikarten, die mit dem Begriff *Sicherheit* verschlagwortet sind. Die Suche erzeugt einen Raum der Dimension 1 durch das Zettelkastenuniversum. D. h. alle Karten sind bereits mit *Sicherheit* verschlagwortet, der Pfad führt also nicht auf neue Begriffe hinaus.

Abb. 9 zeigt die Kollokationen zum Begriff *Sicherheit*. Der Begriff *Sicherheit* steht im Zentrum und es werden Pfade zu allen Schlagwörtern gezeigt, auf denen auch *Sicherheit* steht. Diese Kollokationen erzeugen einen 19-dimensionalen Raum im Zettelkastenuniversum. Es ist mir somit möglich, nicht-offensichtliche Beziehungen zwischen verschiedenen Begriffen zu erkennen und auszuwerten.

Abb. 10 zeigt exemplarisch einen solchen Raum, wobei hier die Kanten beschriftet, also mit ihrer Semantik versehen, sind.

Man kann im elektronischen Zettelkasten die Kollokationen der Stichwörter auch graphisch auswerten. Dazu habe ich ein Skript geschrieben, das zu einem gegebenen Stichwort alle Karteikarten heraussucht und alle weiteren Stichwörter der betreffenden Karteikarten in einer Liste erfasst. Es erzeugt abschließend einen Graphen, der die entsprechenden Kollokationen zeigt. Dies vereinfacht die Auswertung der Beziehungen der einzelnen Karteikarten untereinander.

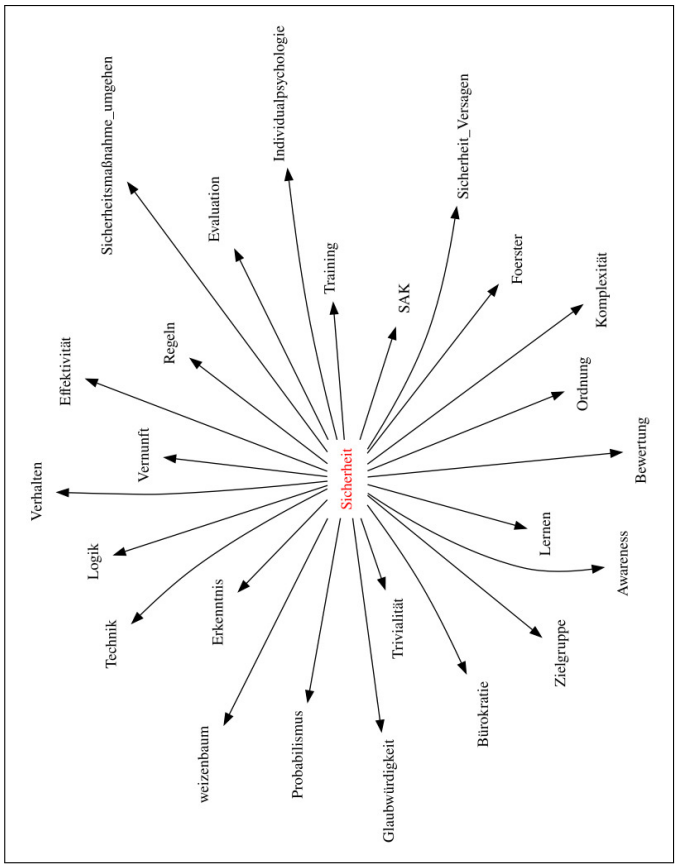

Abbildung 9: Der Graph zeigt die Kollokationen zum Stichwort »Sicherheit«.

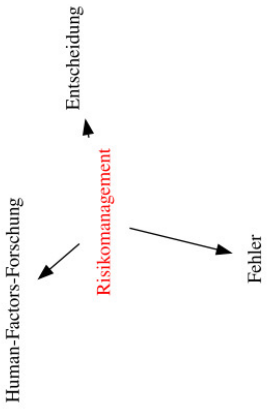

Abbildung 10: Beispiel für ein Netzwerk

3.1 VORTEILE DES ELEKTRONISCHEN ZETTELKASTENS

Gegenüber dem klassischen Zettelkasten hat der elektronische einige Vorteile. So kann eine Karte nicht durch falsche Einordnung verloren gehen, da die Datenbank die Karten sortiert.

Außerdem ist es möglich, die Karteikarten komplett zu durchsuchen, also auch den Textkörper. Dies ist von Vorteil, wenn ein Datum auf einer Karte nicht verschlagwortet wurde. Allerdings muss man hierbei beachten, dass lediglich eine Volltextsuche mit Regulären Ausdrücken über alle Karteikarten gestartet wird. Es ist daher nicht möglich, Karteikarten *semantisch* auszuwerten. Dazu ist nur ein Mensch in der Lage, nicht jedoch ein so primitives Konstrukt wie ein Rechner. Daher ist auch im elektronischen Zettelkasten eine sorgfältige Verschlagwortung erforderlich.

Weiterhin lassen sich Metadaten des Zettelkasten auswerten. So kann bspw. anhand der Zeitstempel überprüft werden, wann welche Karteikarte erstellt oder geändert wurde. Danach lassen sich Listen sortieren oder Cluster erstellen, anhand derer die Epigenese des Zettelkastens untersucht werden kann.

Darüberhinaus spart der elektronische Zettelkasten Tipparbeit, da die Daten per Copy-und-Paste kopiert werden können.

4

WEITERE AUFGABEN

Das Wissen macht uns weder besser, noch glücklicher.

(Heinrich von Kleist)

Der Zettelkasten ist gegenwärtig für Hausarbeiten bzw. als Archiv von Zitaten hervorragend nutzbar. Zur Wissensgenerierung bzw. in der Langzeitarchivierung fehlen aber noch einige Funktionen.

So kann es sinnvoll werden, die Karteikarten weiter auszudifferenzieren und neben »Zitaten« und »Gedanken« weitere Universen zu etablieren.

Ebenso kann es sinnvoll sein, Stichwörter zu hierarchisieren. Allerdings bin ich hier noch unschlüssig, ob und wie sich eine solche Hierarchie umsetzen lässt. Ich kann dies als Baum oder gerichteter Graph tun, wie in Abb. 11 bzw. 12 dargestellt. Der Baum hat den Vor- bzw. Nachteil, das nur genau ein Pfad von der Wurzel zu einem Blatt existiert. Das heißt, ich kann bspw. »Rollenlernen« nur entweder der Sozial- oder der Entwicklungspsychologie zuschlagen. Dies wird schwierig, da verschiedene Themen/Kategorien/Fächer verschiedene Eltern haben können. Im gerichteten Graphen existiert diese Beschränkung nicht, dafür ist er komplexer umzusetzen als ein Baum. Egal wie ich die Hierarchie mathematisch auffasse, zuerst muss sie erst einmal entwickelt werden. Dies ist aber kompliziert, da ich jetzt schon wissen müsste welche Kategorien ich in 10 – 20 Jahren benötige. Daher verzichte ich momentan auf eine Kategorisierung, gehe aber davon aus, das sie mit weiteren Arbeiten sukzessive entstehen wird.

Die Revisionskontrolle bietet noch Entwicklungspotenzial im Bereich Metadatenauswertung. So kann ich bspw. erfassen, welche Kategorien/Stichwörter häufig bearbeitet werden und daraus neue Daten *über den Zettelkasten* generieren.

Eine Art Lesezeichenliste für potenzielle Quellen ist wünschenswert. Oftmals stolpere ich quasi über ein Buch, Journal o. ä., das sich zu lesen lohnt. Da ich in der Regel aber nicht sofort das Werk exzerpieren kann, möchte ich ein Verzeichnis erstellen, in dem es erfasst und »zwischengespeichert« wird. Dazu werden die verfügbaren bibliographischen Daten erfasst und um Stichwörter und eine Kurzzusammenfassung ergänzt. Momentan besteht die Bibliographie nur aus Werken die auch verarbeitet wurden. Mit dieser Lesezeichenliste erweitert sie sich um Werke die noch verarbeitet werden. Der Zettelkasten antizipiert also schon die zukünftige Wissensentwicklung.

Die graphischen Funktionen können optimiert werden. Eventuell bieten sich hierzu andere Techniken als automatisch mit GraphViz generierte Graphen an.

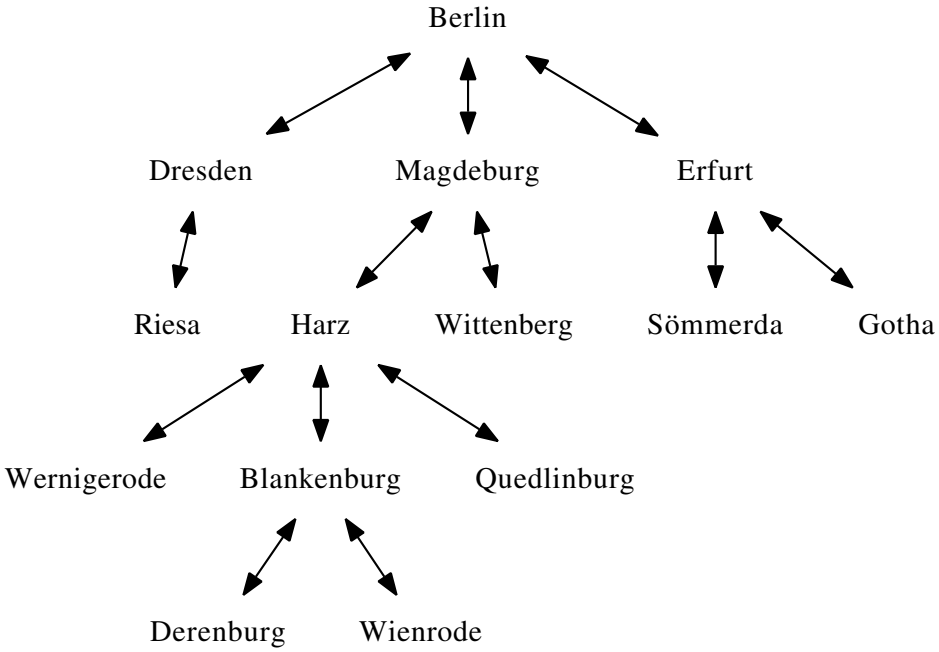

Abbildung 11: Hierarchie als Baum

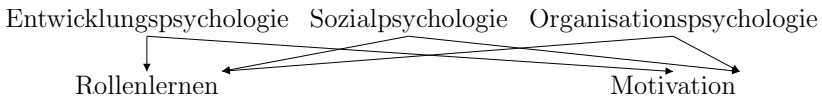

Abbildung 12: Hierarchie als Graph

Die generierten Graphen können erweitert werden. Gegenwärtig sind die Graphen ein-dimensional, d. h. es gibt von der Wurzel (Stichwort) jeweils nur einen Sprung auf ein anderes Stichwort. Theoretisch ist es möglich, alle Stichwörter des Zettelkastens in einem Graphen darzustellen. Praktisch scheitert dies aber am Speicherbedarf, GraphViz hat mit einem Speicherzuordnungsfehler auf einer Maschine mit 8GB RAM abgebrochen. Eventuell gibt es geeignetere Techniken, um eine »Landkarte« des Zettelkastens zu erstellen.

Außerdem wäre es sinnvoll, einen solchen Graphen auch für die Bibliographie zu erstellen, um zu zeigen, wie verschiedene Werke in Beziehung zueinander stehen.

Die Graphen könnten ebenso als semantisches Netzwerk dargestellt werden. Das heißt, das bspw. die Kanten benannt oder gewichtet werden oder im Knoten dargestellt wird, wie häufig ein Stichwort vorkommt. Auf der Startseite des Zettelkastens wird dies teilweise über die metrische Stichwortwolke dargestellt. Je häufiger ein Stichwort ist, desto größer wird es dargestellt.

Die Datenbank sollte um die Handhabung von Binärdateien (BLOBs[1]) erweitert werden. Dadurch wäre es möglich bspw. Bilder, PDFs oder auch OGG-Dateien abzulegen.

Die rudimentären Kollaborationsmechanismen müssen ausgebaut werden, so das effektiv mehrere Benutzer mit dem Zettelkasten arbeiten können. Dazu muss bspw. der Benutzername des anlegenden Benutzers auf einer Karteikarte hinterlegt werden. Ebenso ist es notwendig, ggf. Zugriffsrechte auf Karten durchzusetzen.

Eventuell kann eine integrierte Verwaltung meiner Artikel im Zettelkasten sinnvoll werden. Mit wachsender Zahl an Artikeln wird es schwierig, den Überblick zu behalten. Daher kann es notwendig sein, fertige Artikel in der Datenbank abzulegen und sie so zumindest in Teilen durchsuchbar zu machen.

Einige Eyecandy[2]-Funktionen für den Editor bzw. die Listen wären sinnvoll. Beispielsweise eine Liste, die alle Karten anzeigt und bestimmte Karten dabei zusammen- bzw. ausklappt. Dazu wäre JavaScript o.ä. erforderlich. Ebenso könnte ich im Editor Klickfelder für Fett, Kursiv, Links etc. einbauen.

Eigentlich wäre es sinnvoll, einen öffentlichen Zettelkasten ähnlich der Wikipedia oder Bibsonomy[3] zu führen. Bibsonomy ist eine Webseite, auf der man eine Bibliographie und Lesezeichenliste online führen kann. Die Einträge werden verschlagwortet und können anhand der Schlagworte gesucht werden. Da Bibsonomy von mehreren Benutzern benutzt wird, entsteht so eine *Social Tagging Cloud*, eine soziale Schlagwortwolke. Diese bietet die Möglichkeit, in das Literaturuniversum anderer Benutzer einzusteigen und so weitere Anregungen zu erhalten. Ähnliches wäre auch im Rahmen des Zettelkastens möglich, so das jeder Benutzer seine Zitate online sammelt und verschlagwortet. Der gesamte Zettelkasten würde dann unterschiedliche Benutzer, Sichtweisen und Zitatsammlungen vereinen und könnte damit wiederum neue Ideen generieren. Anders als bei Wikipedia sind die Einträge eines Benutzers nur vom Benutzer selbst editierbar, damit werden

1 Binary Large Object
2 Eyecandy: Hübsch anzusehen, aber nicht wirklich notwendig.
3 http://www.bibsonomy.org/

Relevanzdiskussionen und Edit-Wars[4] vermieden. Problematisch ist hierbei aber das Urheberrecht. Ich gehe davon aus, das eine solche Zitatsammlung schnellstmöglich aus dem Netz geklagt werden wird.

4.1 SICHERHEITSPROBLEME

Ein elektronisch geführter Zettelkasten birgt andere Risiken als ein Zettelkasten aus Holz und Papier.

Von Vorteil ist hierbei, das sich der elektronische Zettelkasten leicht replizieren und bequem in eine Datensicherungsstrategie integrieren lässt. So ist es ohne weiteres möglich, funktionierende Kopien auf dutzenden Rechnern vorzuhalten.

Daher fertige ich regelmäßige Backups des Zettelkasten an, die ich auf mehrere Datenträger an verschiedenen Orten verteile.

Problematisch ist jedoch, Daten geheimzuhalten. Dies mag bei Zitaten aus öffentlichen Werken nicht besonders wichtig scheinen, jedoch verarbeite ich hier auch Daten aus Projekten mit erhöhtem Schutzbedarf bzw. vertraglich zugesicherter Geheimhaltung.

Daher ist es für mich überlebenswichtig, die entsprechenden Daten zu schützen. Dies erreiche ich teilweise durch anonymisierte Erfassung, die aber (noch) nicht immer möglich ist. Daher gliedere ich sensible Daten in einen unabhängigen, projektbezogenen Zettelkasten aus. Dessen Datenbestände schütze ich durch Verschlüsselung des Dateisystems (vgl. Schumacher, 2006) sowie Auslagerung auf externe Datenträger (Datensticks). PostgreSQL bietet mit seiner Tablespace-Funktion die Möglichkeit, den physikalischen Speicherort von Relationen nahezu beliebig zu wählen. Dies lässt sich ausnutzen, um einzelne Relationen auszulagern, bspw. in verschlüsselte Verzeichnisse oder auf Wechseldatenträger.

Eine besondere Gefahr ist die mögliche Manipulation von Daten. Ein Angreifer könnte in den Server einbrechen und Daten in der Datenbank verändern, was die Integrität des Zettelkastens unterminiert.

Daher verwende ich Prüfsummen und Signaturen (vgl. Schumacher, 2004) um die Integrität der relevanten Datensätze zu sichern. Die Signatur fungiert dabei als eine Art »elektronische Unterschrift«, d. h. sollte der Datensatz geändert werden, passt die ursprüngliche Signatur nicht mehr und die Manipulation fällt auf.

4 http://de.wikipedia.org/wiki/Wikipedia:Edit-War

5

CODA

The city's central computer told you? R2D2, you know better than to trust a strange computer!

(C3PO in »Star Wars«)

Der Zettelkasten ist seit Jahrhunderten ein geeignetes Mittel um Daten zu verwalten. Bei disziplinierter Führung entsteht ein System, mit dem man kommunizieren kann, um Artikel zu erstellen. Er bildet somit eine sehr gute Basis, um eigene Forschungsvorhaben erst zu begleiten und später sogar zu generieren.

Der elektronische Zettelkasten greift dieses System auf und passt es an die verfügbaren Techniken an. Im Grundprinzip bleibt er ein Zettelkasten, wie in Luhmann (1993) beschrieben.

Allerdings ermöglicht die Rechentechnik den Einsatz neuer Techniken, wie der oben beschriebenen Kollokationsgraphen, der Volltextsuche und der Revisionskontrolle.

Während die ersten beiden Punkte nur Funktionen des klassischen Zettelkastens vereinfachen, ermöglicht die Revisionskontolle mehr.

Die Revisionen des Zettelkastens kondensieren quasi seine Synaptogenese und ermöglichen es so, die Entwicklungsgeschichte nachzuverfolgen. Daraus lassen sich Rückschlüsse auf meine Arbeit und nicht zuletzt die »Persönlichkeit« des Zettelkastens ziehen.

Dabei ist natürlich zu beachten, dass der Zettelkasten selbst kein neues Wissen generiert, sondern nur Daten vorhält. Die Wissensgenese erfolgt nur durch mich.

Ansonsten vereinfacht der elektronische Zettelkasten die Arbeit an neuen Artikeln ungemein. Zahlreiche algorithmisierbare Schritte werden automatisiert und von Skripten erledigt.

6

LITERATUR

Luhmann, N. (1993). Kommunikation mit Zettelkästen. Ein Erfahrungsbericht. In A. Kieserling (Hrsg.), *Universität als Milieu* (1. Ausgabe, S. 53–61). Haux.

Mietzel, G. (2001). *Pädagogische Psychologie des Lernens und Lehrens* (6., korrigierte Auflage). Hogrefe.

Niegemann, H. M., Hessel, S., Hochscheid-Mauel, D., Aslanski, K., Deimann, M., & Kreuzberger, G. (2003). *Kompendium E-learning* (1. Aufl.). Springer.

Reischmann, J. (2002). *Weiterbildungs-Evaluation: Lernerfolge messbar machen* (1. Auflage). Luchterhand.

Schumacher, S. (2004). Einführung in kryptographische Methoden. Verfügbar 19. April 2004 unter http://www.cryptomancer.de/21c3/21c3-kryptograhie-paper.pdf

Schumacher, S. (2006). Verschlüsselte Dateisysteme für NetBSD. *UpTimes*, 25–31. Verfügbar 7. November 2009 unter http://kaishakunin.com/publ/guug-uptimes-cgd_cfs.pdf

Spitzer, M. (2006). *Lernen: Gehirnforschung und die Schule des Lebens* (6. Aufl.). Spektrum Akademischer Verlag.