

PKM -Personal Knowledge Management

Johannes Buchner und Daniel Fehrenbach

2. Februar 2002

Inhaltsverzeichnis

I	Theoretischer Teil	2
1	Einleitung	2
1.1	Informationsmanagement Anfang des 21. Jahrhunderts	2
1.2	generelle Vorteile elektronischer Informationsspeicherung	2
1.3	Vorteile elektronischer Informationsverarbeitung	3
1.4	Forschungsziel dieses Projektes	4
2	Methodik / Vorgehensweise	4
2.1	Voraussetzungen	4
2.2	Die Unix-Philosophie	4
2.3	Die Open-Source-Bewegung	4
2.4	Internet und Sourceforge.net	5
2.5	Einfluß auf PKM	5
2.6	konkrete Vorgehensweise	5
3	elektronische Mitschriften	6
3.1	Dateiformat	6
3.2	Eingabe-Tools	7
3.3	Versionskontrolle	8
4	Strukturieren und Ordnen von Informationen	8
4.1	Probleme der strukturierten Informationsspeicherung	8
4.2	Konzept einer relationalen MySQL-Datenbank	9
5	Ausblick	9
5.1	Das PKM von übermorgen	10
5.2	Das Wissen der Menschheit zugänglich machen - eine Vision für die Zukunft	10
II	Praktischer Teil	11
6	Integration von Vim, L^AT_EX, DocBook, CVS für elektronische Mitschriften	11
6.1	L ^A T _E X-Mappings	11
6.2	DocBook-XML	11
6.3	XML und Vokabel-Tags und Mappings	11
6.4	CVS-Mappings	12

7 Die PKM-Dokumentendatenbank	12
7.1 Mysql und PHPMyAdmin	12
7.2 Eintragen von Dokumenten in die PKM-Datenbank	12
7.3 Zugriff auf die Datenbank	13
8 Fazit zum jetzigen “Stand der Dinge”	14
9 mögliche Erweiterungen - die nächsten Schritte bei PKM	14
9.1 Zugriffsmöglichkeiten auf nicht-persönliche Daten	14
9.2 XML bringt neue Möglichkeiten	14
9.3 Integration von nicht-elektronischen Dokumenten	15
9.4 Anbindung an PIM	15

Teil I

Theoretischer Teil

1 Einleitung

1.1 Informationsmanagement Anfang des 21. Jahrhunderts

In einem Menschenleben ist man ständig mit Informationen konfrontiert. Insbesondere in den Lebensabschnitten von Schule und Studium (zwischen welchen beiden die Autoren als Zivildienstleistende stehen) kommt es darauf an, Wissen zu erwerben und Informationen zu recherchieren und diese in irgendeiner Form abzulegen (und sei es nur im Gedächtnis!). Zur dauerhaften Informationsspeicherung und -weitergabe wird in der heutigen Gesellschaft meist noch auf das Medium Papier zurückgegriffen: in der Schule beispielsweise lernt man aus Schulbüchern und schreibt in Hefte. Interessiert man sich für ein bestimmtes Thema, so informiert man sich zunächst in Zeitschriften oder Sachbüchern.

Das Internet hat in den letzten Jahren als Informationsmedium einen rapiden Aufstieg erlebt. Nichtsdestotrotz ist die Mehrheit der heute lebenden Menschen nicht mit elektronischen Medien großgeworden, weswegen zum Teil sogar eine Ablehnung von Internet und Computertechnologie zu beobachten ist. Dies lässt sich unter anderem auch daran ablesen, dass das Internet zwar zur schnellen Informationsrecherche immer mehr Zuspruch findet, allerdings werden wichtige Dinge, die man dauerhaft aufbewahren will, oft ausgedruckt, so beispielsweise E-Mails, Datenblätter, Tabellen etc. In den elektronischen Medien wird also recherchiert, wenn es an die Informationsspeicherung und -archivierung geht wird also oft noch auf das Medium Papier zurückgegriffen.

1.2 generelle Vorteile elektronischer Informationsspeicherung

Der Computer bietet die Möglichkeit, jede Art von Information in binärer Form auf Datenträgern zu speichern. Im Moment ist dies beispielsweise bei Text, Bildern, Audio und Video möglich, aber prinzipiell gilt dies auch für andere Sinneswahrnehmungen, wenn ein entsprechendes “Aufnahmegerät”, Dateiformat, und “Wiedergabegerät” existiert (bei Ton wäre dies z.B. “Mikrofon + Soundkarte”, “wav-Format”, “Lautsprecher + Soundkarte”). Ist eine Information erst einmal digital erfasst und als Datenstrom, der im Prinzip nur aus “Nullen und Einsen” besteht, in einer Datei abgelegt, so bietet dies prinzipiell entscheidende Vorteile gegenüber der physikalischen Version der Information:

- **verlustfreie Kopien:** von jeder Datei lassen sich beliebig viele Kopien erstellen, ohne dass dadurch ein Verlust der Darstellungsqualität befürchtet werden muss (wie etwa beim Kopieren eines Bildes durch ein Kopiergerät!)
- **einfache Vervielfältigbarkeit:** es ist kein großer Aufwand, eine Kopie einer Datei zu erstellen (ein elektronisches Adressbuch beispielsweise lässt sich durch Anhängen an eine Email weitergeben, eines auf Papier nur durch aufwändiges Kopieren am Kopierer)

- **universelle Speichermöglichkeit:** Dateien lassen sich ohne Probleme auf den verschiedensten Speichermedien ablegen. Durch moderne Speichertechnologie kann sehr viel Wissen auf sehr kleinem Raum gespeichert werden, unter anderem Dank moderner Kompressionsverfahren (elektronische Daten lassen sich anhand mathematischer Algorithmen verlustfrei komprimieren, je nach Datenmaterial allerdings unterschiedlich gut).
- **problemloser Datentransport:** Dateien lassen sich über elektronische Leitungen oder per Funk versenden und empfangen, z.B. über das Internet oder das Mobiltelefon
- **prinzipiell unbegrenzte "Haltbarkeit":** elektronisch gespeicherte Informationen sind nur an die physikalische Haltbarkeit ihres Mediums gebunden, sind aber durch die Möglichkeit der verlustfreien Kopie prinzipiell unbegrenzt "haltbar". Dazu ist es allerdings notwendig, dass ältere Dateien auf neue Speichermedien umkopiert werden, da veraltete Medien auf modernen Computern meist nicht mehr lesbar sind. Neue Medien stellen dann allerdings oft ein vielfaches der Kapazität zur Verfügung (Für die Daten einer CD-ROM bräuchte man ca. 462 Disketten!!). Bei Papier hingegen lässt sich der physikalische Alterungsprozess irgendwann kaum aufhalten oder umgehen.
- **Zugänglichkeit/Erreichbarkeit:** elektronische Informationen können durch technische Hilfsmittel einem größeren Personenkreis zugänglich gemacht werden, auch behinderten Menschen. Blinde können z.B. per Braille-Zeile und Sprachausgabe Zugang zu jeglicher Art von geschriebener elektronisch vorhandener Information erhalten (z.B. auch zum Internet!).¹

Zusätzlich ergeben sich noch viele andere Vorteile, die aber von der Art der Information abhängen (z.B. Bilder: nachträgliche Bildbearbeitung mit Möglichkeiten wie "Kontrast erhöhen", "Helligkeit nachregeln", "Filter xy anwenden" etc.).

1.3 Vorteile elektronischer Informationsverarbeitung

Auch wenn es an das Strukturieren, Sortieren oder Vernetzen von Wissen geht bieten elektronisch abgelegte Informationen viele Vorteile:

- **Wissensmenge:** durch die Speicherkapazität moderner Rechner kann man problemlos sehr viel Wissen speichern, z.B. ganze Lexika wie den Brockhaus, Wörterbücher, Atlanten, andere Datenbanken etc.. Dadurch steht bei der Arbeit an Computern/Laptops immer eine riesige digitale Bibliothek zur Verfügung, die genutzt werden kann
- **Vernetzen/Hypertext:** Dieses Wissen kann viel besser verknüpft werden: Hypertext bietet Vorteile gegenüber Papier, man kann auftauchende neue Begriffe direkt im Lexikon nachschlagen etc. Hypertext bedeutet, daß Text nicht einfach sequentiell (wie z.B. in einem Buch, bzw. auf Papier eigentlich immer) abgelegt wird, sondern durch Links, also Querverweise, miteinander verknüpft. Dadurch werden Beziehungen klarer, es wird sofort deutlich, wenn zu einem Thema noch mehr Information zur Verfügung steht, es kann direkt auf zitierte oder genannte Dokumente verwiesen werden u.ä. Das Hypertext-Konzept ist zwar schon älter, aber der Durchbruch kam erst durch den Computer, vor allem durch das auf HTML (Hypertext Markup Language) basierende Internet.
- **Geschwindigkeit:** man kann die Geschwindigkeit des Rechners nutzen, der problemlos stupide Dinge wie "alle Dateien nach Stichwort xy durchsuchen" ausführt, und muss sich also nicht damit beschäftigen, eine alphabetische geordnete Liste nach dem richtigen Begriff zu durchsuchen (wozu man gezwungen ist, wenn man einen Begriff in einem Lexikon in Buchform nachschauen möchte...)
- **Unabhängigkeit:** Internet bietet die Möglichkeit, dass Information unabhängig von Ort und Zeit zugänglich ist, d.h. man kann auch nachts noch Anfragen an die "digitale Bibliothek" stellen, ohne an Öffnungszeiten gebunden zu sein. Zusätzlich ist auch der physikalische Ort des Netzzuganges unerheblich, man kann sich also z.B. mit dem Laptop in den Wald setzen und per Handy-Funknetz einen

¹Einer der Autoren leistet seinen Zivildienst an einer Universität ab, wo die Betreuung von blinden StudentInnen zu seinen Aufgaben gehört und die Nutzung der Informationstechnologie beispielsweise einen Zugang zum Internet für Blinde ermöglicht !

Begriff im Lexikon nachschlagen (ohne Zweifel ein Vorteil verglichen mit kiloschweren gebundenen Lexika-Ausgaben !).

- **Suchmöglichkeiten:** elektronische Daten bieten erweiterte Suchmöglichkeiten: nach Stichwörtern oder, wenn das nicht reicht, nach regulären Ausdrücken. Reguläre Ausdrücke (Regular Expressions) gehören zu formalen Sprachen, ein Teilgebiet der Informatik. Mit regulären Ausdrücken lassen sich die Merkmale des gesuchten Textteils formal beschreiben, sucht man z.B. nach einer deutschen Mobilfunk-Telefonnummer, kann man dies mit dem regulären Ausdruck `(0|+49)1[67][0-9][[:space:]]?[0-9]{7}` tun (eine deutsche Mobilfunknummer fängt mit einer Null für Vorwahl oder +49 für Deutschland an, geht mit einer Zahl zwischen 160 und 179 weiter, also zuerst eine 1, dann eine 6 oder eine 7, dann eine beliebige Ziffer; je nach Notation folgt dann ein Bindestrich, ein Leerzeichen, ein Schrägstrich oder aber auch gar kein Trennzeichen, dann folgen 7 beliebige Ziffern). Das Beispiel läßt die Möglichkeiten regulärer Ausdrücke nur erahnen, es sind auch deutlich komplexere Suchmuster möglich.
- **Datenbanken:** Datenbanken bieten bei der Informationsspeicherung viele Möglichkeiten, die weit über das bloße Ablegen einer Datei in einer Verzeichnisstruktur hinausgehen (siehe 4.2).

1.4 Forschungsziel dieses Projektes

In dieser Arbeit geht es darum herauszufinden, wie man die Vorteile, die elektronische Datenverarbeitung bei der Speicherung und Strukturierung von Wissen bieten kann, konkret für sich nutzen kann. Für mögliche Probleme, die ein digitales Informations- und Wissens-Management mit sich bringen könnte, sollen Lösungsansätze aufgezeigt werden. Insgesamt soll deutlich werden, wie ein Mensch die Vorteile der neuen Kulturtechnik der elektronischen Datenverarbeitung für sich und seine persönliche Lern- und Wissenssituation nutzen kann (z.B. als StudentIn).

2 Methodik / Vorgehensweise

2.1 Voraussetzungen

Da die Autoren einen technisch-naturwissenschaftlichen Hintergrund besitzen, sich also schon mit dem Aufbau, der Programmierung und der Administration von Computern und deren Betriebssystemen beschäftigt haben, bestand das Forschungsprojekt größtenteils aus dem Sammeln von Informationen und dem Zusammentragen dessen, was bereits in irgendeiner Form vorhanden war und für das Projekt nützlich sein konnte. In diesem Zusammenhang sind die "Unix-Philosophie" und der "Open-Source-Gedanke" von zentraler Bedeutung.

2.2 Die Unix-Philosophie

Als in den 60er Jahren Unix geschaffen wurde, hatten seine Schöpfer den Anspruch, ein Betriebssystem mit einem sauberen Konzept zu designen. "Saubere" bedeutet in diesem Zusammenhang erweiterbar, mit der Anlagen zu vielem, einheitlich, d.h. möglichst wenig Spezialfälle, aber vor allem wiederverwendbar. Letzteres um zu verhindern, daß ähnliche Dinge immer wieder neu programmiert werden müssen. Stattdessen bemühte man sich darum, Interaktionsmöglichkeiten zwischen Programmen zu schaffen. Es sollte viele kleine Programme geben, die sehr wenig Funktionalität haben, diese aber auch wirklich stabil beherrschen. Diese sollten über eine saubere, d.h. klar definierte und flexible Schnittstelle zusammenarbeiten und immer so wieder neu kombinierbar sein (Stichwort Modularität), was eine enorme Flexibilität und Mächtigkeit zur Folge hat.

2.3 Die Open-Source-Bewegung

In den 80er Jahren entstand die Idee der "Freien Software", welche inzwischen wegen der Doppeldeutigkeit des Wortes "Frei" in "Open Source" umbenannt wurde. "Frei" soll hier nämlich nicht "kostenlos" sondern "besitzerlos" bedeuten!

Bis dahin entstand Software auf folgendem Weg: eine Firma stellt Programmierer ein, die ein Programm schreiben. Damit sie ihre Leute bezahlen kann, muss die Firma Umsatz machen, den sie im Normalfall durch den Verkauf des Programmes erzielt ("kommerzielle Software"). Das Programm wird nur in einer binären Version herausgegeben, in der man es zwar benutzen kann, nicht es nicht möglich ist zu erkennen, wie es funktioniert. Der Quellcode bleibt als Geheimnis in der Firma, so wie Baupläne für Autos oder Verfahren auch beim Autohersteller bleiben. Sonst könnte jederman den Quellcode ein bißchen abändern und ein Konkurrenzprodukt verkaufen, ohne die Forschungsarbeit geleistet zu haben, und die Firma ginge pleite. Wenn irgendjemand etwas ähnliches Programmieren möchte, muß er ganz von vorne anfangen.

Nun gab es aber auch Menschen, die ihre Arbeit der Gemeinschaft zur Verfügung stellten, ohne Geld dafür zu verlangen. Gegen die Strategie des "Quellcodes als Firmengeheimnis" setzt die Open Source-Bewegung die Regel, alle Quellen offenzulegen, und es niemandem zu verbieten, sie zu modifizieren, solange dieser seine Veränderungen ebenfalls als "Open-Source" zur Verfügung stellt. Letzteres sichert die "GNU² Public License" (GPL) ab, unter der auch die für PKM entwickelten Skripte, Config-Files etc. stehen. Dadurch kann jeder Programmierer auf einen riesigen Fundus an freier Software und deren Quellcode zugreifen und braucht "das Rad nicht mehr jedes Mal neu erfinden". Zusätzlich dazu ist freie Software meistens außerordentlich gut dokumentiert, im Internet finden man völlig kostenlos sowohl Tutorials für Einsteiger als auch Referenzartikel für Profis und Entwickler. Letzters hat bei der Informationssammlung für PKM natürlich sehr geholfen!

2.4 Internet und Sourceforge.net

Für die Open-Source-Entwicklung stellt das Internet **das** zentrale Kommunikationsmedium dar. E-Mail (insbesondere Mailinglisten zu bestimmten Themen) und elektronische Diskussionsforen ermöglichen insbesondere Projekte, die aus mehreren Entwicklern bestehen, die nicht alle in der selben Region/Land/Kontinent leben und deshalb auf das Internet als Kommunikationsmedium angewiesen sind.

Sourceforge ist eine Internet-Plattform für Open-Source-Entwickler, die verschiedene Web-basierte Dienstleistungen wie projektbezogene Mailing-Listen, Diskussionsforen, CVS-Server etc. kostenlos zur Verfügung stellt. Für unser Projekt war vor allem die Plattform in Form einer eigenen Webseite³ wichtig, um in Zukunft Kontakt zu anderen Interessierten Entwicklern zu bekommen. Außerdem "hostet" (d.h. beherbergt) Sourceforge eine große Anzahl an freien Software-Projekten und bietet auch gute Dokumentationen zu den verwendeten Werkzeugen wie CVS.

2.5 Einfluß auf PKM

Gemäß den gerade beschriebenen Konzepten und Ideen "erfindet" PKM weder einen Editor, ein Dateiformat o.ä. neu, sondern verwendet vorhandenes, um es zu etwas Neuem zu verknüpfen. Da alle verwendete Software frei ist, ist dies auch legal und entspricht genau der Open-Source- und Unix-Philosophie von Wiederverwendbarkeit und dem gemeinsamen Entwickeln, wovon schließlich wieder alle profitieren. Die im Rahmen des PKM-Projektes entwickelten Programme, Skripte, Konfigurations-Dateien etc. stehen deshalb auch selbst unter der GPL[11], außerdem werden auf der Homepage des Projektes (<http://pkm.sourceforge.net>) andere dazu aufgefordert, an PKM mitzuarbeiten und ihre Ideen beizutragen.

2.6 konkrete Vorgehensweise

Die Forschungsarbeit bei PKM bestand also vor allem aus Internetrecherchen, aber auch im Lesen von Büchern zu verschiedenen Themen, um einen Überblick über die Möglichkeiten der verschiedenen Werkzeuge zu erhalten.

Bei der Suche nach geeigneten Werkzeugen für PKM haben sich zwei Bereiche herauskristallisiert: "Tools (d.h. Werkzeuge) zur Unterstützung elektronischer Mitschriften" und "Tools zum Strukturieren und Ordnen von Informationen".

²GNU steht für "GNU is Not Unix" und stellt somit ein rekursives Akronym dar

³<http://pkm.sourceforge.net>

3 elektronische Mitschriften

Wenn man sein persönliches Wissens- und Informations-Management komplett auf elektronische Weise erledigen will, dann muss man sich natürlich auch angewöhnen, Mitschriften z.B. von Vorlesungen, Seminaren, Vorträgen etc. statt auf Papier mit dem Computer zu erledigen. Die hierzu getätigten Überlegungen sind im nächsten Abschnitt zusammengefasst.

3.1 Dateiformat

3.1.1 Anforderungen

An das Format, in dem Daten wie Vorlesungsmitschriften, Buch-Excerpte etc. in PKM abgelegt werden, werden folgende Anforderungen gestellt:

- Es muß Strukturierungsmöglichkeiten bieten, die für PKM nützlich sind. Dies sind in erster Linie logische Abschnitte zur Strukturierung, verschiedene Text-Attribute wie **fett** oder *kursiv* und Hyperlinks zur Vernetzung von Wissen
- Es müssen Exportfilter existieren, um die PKM-Daten leicht in andere Formate überführen zu können, und sei es nur zur Darstellung
- Es muß frei und open-source sein, da das ganze Projekt unter GPL steht und andere dazu aufgerufen sind, an der Verbesserung mitzuarbeiten
- Es muß möglichst plattformunabhängig sein, d.h. auf vielen Plattformen zur Verfügung stehen, auch um in Zukunft die Benutzbarkeit auf anderen Endgeräten wie Mobiltelefon oder Organizer zu gewährleisten
- Es sollte erweiterbar sein, um auch für speziellere Anwendungen genügend Flexibilität zu bieten

Die obigen Anforderungen werden von den folgenden zwei Formaten erfüllt:

3.1.2 L^AT_EX

L^AT_EX ist ein Satzsystem. Es beherrscht die Regeln der Buchdruckerkunst, d.h. Strukturmerkmale von Büchern, Artikeln u.ä sind vordefiniert. Außerdem kennt L^AT_EX strukturegebende Elemente wie Abschnitte, Überschriften, Listen, Aufzählungen uvm. und “weiß”, wie diese auszusehen haben. Man gibt lediglich ein kurzes Zeichen, wenn ein neuer Sinnabschnitt oder ein neuer Unterabschnitt beginnt, und L^AT_EX kümmert sich um Nummerierung, Formatierung und bietet sofort eine grobe Visualisierung. Es existieren zahlreiche Exportfilter nach HTML, XML uvm. L^AT_EX basiert auf ASCII (d.h. “American Standard Code for Information Interchange”, eine einfache, grundlegende Norm, um Texte mit Computern darzustellen), also kann man L^AT_EX-Code in dem Editor seiner Wahl erstellen, und sich den Output hinterher anschauen. ASCII ist der kleinste gemeinsame Nenner aller Plattformen. L^AT_EX ist allerdings nicht erweiterbar.⁴

3.1.3 XML

XML steht für “eXtensible Markup Language”. Es wird oft als Nachfolger von HTML, worin zur Zeit fast alle Dokumente im WWW verfaßt sind, bezeichnet; es ist aber viel mehr. Es ist eine Meta-Sprache⁵, in der man Information völlig losgelöst von der Darstellung strukturiert ablegen kann. Das hat es mit L^AT_EX gemein. Es ist aber außerdem noch erweiterbar (extensible), was bedeutet: während bei HTML oder L^AT_EX die Strukturelemente vorgegeben sind (wie z.B. für Überschrift oder Kapitelname), kann man sie sich in XML frei definieren und so an seine Aufgabe anpassen. Immer öfter wird inzwischen auf XML als Dateiformat zurückgegriffen, und so werden die zahlreichen Konverter, die ein Format in ein anderes

⁴Übrigens: diese Arbeit wurde in L^AT_EX mit Hilfe von L^AX erstellt!

⁵d.h. man kann durch XML andere Markup-Sprachen definieren, z.B. lässt sich HTML komplett in XML definieren (es nennt sich dann “XHTML”)

verwandeln, eines Tages wegfallen, wenn dann fast alles in XML abgelegt ist. XML ist dabei frei, basiert auf ASCII, ist dadurch also plattformunabhängig, und gut dokumentiert durch das W3C⁶[6]. XML eignet sich für PKM, weil es PKM um das Ablegen von Daten und die Anpassung an die eigenen Bedürfnisse (das P in PKM steht ja immerhin für Personal !) geht.

3.1.4 Ergebnis

Langfristig ist XML das Format der Wahl für PKM, da es erweiterbar, allgemein und deswegen der Standard von morgen ist.

3.2 Eingabe-Tools

Ein Problem des elektronischen Mitschreibens ist, daß dies bei vielen Menschen langsamer geht als eine Mitschrift von Hand. Daher müssen alle Möglichkeiten, die Geschwindigkeit der Texteingabe zu erhöhen, ausgeschöpft werden. Die meisten Textverarbeitungssysteme arbeiten nach dem WYSIWYG-Prinzip. WYSIWYG steht für "What You See Is What You Get" und bedeutet, daß man am Bildschirm bereits sieht, wie später der Output am Drucker aussehen wird. Man hat alle Freiheiten, das Layout und die Formatierung selbst zu bestimmen, damit leider aber sogar die Pflicht, dies zu tun. In WYSIWYG-Systemen wie Microsoft-Word verbringt man in der Regel den größten Teil der Zeit mit Formatierung und den geringen mit der Textproduktion. Das WYSIWYG-Prinzip ist daher eher ungeeignet für PKM. Doch es gibt Alternativen:

3.2.1 Lyx und WYGIWYM

LyX ist auch ein Textverarbeitungssystem. Es nennt das Prinzip, nach dem es arbeitet (nicht gerade bescheiden) WYGIWYM, d.h. "What You Get Is What You Mean". Das bedeutet: Lyx kümmert sich um die Formatierung und das Layout selbst, der/die BenutzerIn muß nur abstrakt die Struktur vorgeben. Wie auch im Abschnitt vi/vim (siehe 3.2.2) ist dies zuerst einmal weniger intuitiv und bedarf Vorwissen und Übung, um benutzt werden zu können. Ist beides gegeben, läßt LyX hohe Arbeitsgeschwindigkeit durch fast völlige Konzentration auf Inhaltliches zu. Aus der Struktur, die der User vorgibt, erstellt LyX ein Inhaltsverzeichnis, mittels dessen man sich leicht einen Überblick verschaffen kann, sei es beim Editieren oder beim Nachschlagen Monate später. LyX kann in zahlreiche Formate exportieren, u.a. DocBook, also XML (siehe 3.1.3), und L^AT_EX (siehe 3.1.2). Außerdem ist Lyx erweiterbar und konfigurierbar: für komplexere Navigation innerhalb von Texten kann man die Emacs-Tastenbelegung einstellen, außerdem existiert mit dem "lyxserver" eine Möglichkeit zur Kommunikation mit externen Programmen⁷ (siehe 6.1.3)

3.2.2 vim/vi

Vi galt früher als **der** Standard-Editor unter Unix, war allerdings kommerziell. Heute gibt es eine große Vielfalt von Vi-Nachbauten, großteils wesentlich mächtiger als das Original, aber alle auf der gleichen Idee beruhend. "Vim" ist einer derjenigen mit den meisten Erweiterungen und Features. Hier nun eine Erläuterung des vi-Editier-Konzeptes:

In anderen Editoren erhält man beim Druck einer Buchstabentaste eben diesen Buchstaben, bei einer Zahlentaste eben diese Zahl. Um im Text zu navigieren, muß also entweder auf die Maus und Pull-Down-Menüs (Microsoft Word, Windows Notepad) oder auf mächtige aber komplizierte und unhandliche Tastenkombinationen mit Strg, Shift und Alt zurückgegriffen werden (z.B. GNU Emacs). Ersteres bietet zwar den Vorteil, daß man ohne großes Vorwissen und Übung auf diese Weise editieren kann, hat aber große Nachteile für die Geschwindigkeit. Auf diese kommt es in der Regel auch nicht so sehr an wie bei PKM. In Vi gibt es zwei unterschiedliche Modi: Im Insertmode wird Text eingegeben. Drückt man eine Buchstabentaste, erhält man diesen Buchstaben. Im Commandmode kann im Text navigieren, diesen verändern, kopieren, eingefügen, löschen, formatieren usw. Hierzu steht dann die gesamte Tastatur zur Verfügung, was einerseits zur Folge hat, daß der Schreibende seine Finger stets, gemäß der 10-Fingersystems, auf der

⁶"World Wide Web Consortium", eine internationale Institution, die sich um Standardisierung im WWW bemüht

⁷Mittelfristig könnte LyX dadurch zum Eingabemodul für PKM werden.

mittleren Buchstaben-Zeile liegen lassen kann, andererseits durch das große Angebot an zur Verfügung stehenden Tasten sehr komplexe und mächtige und dennoch intuitive Editierfunktionen möglich sind. Beides führt mit ein bißchen Übung zu einer sehr hohen Geschwindigkeit beim Editieren. Vi und in noch höherem Maße vim lassen sich durch Tastaturlistings erweitern. Dabei wird einer bestimmten Tastenkombination eine benutzerdefinierte Funktion zugewiesen, was sowohl für den Insert- als auch für den Commandmode möglich ist. Es lassen sich komplexe Aufgaben bewältigen, externe Programme oder Filter auf Textteile anwenden etc., was vim eine sehr große Flexibilität und Erweiterbarkeit beschert. Dadurch lässt sich vim sehr einfach an die eigenen Bedürfnisse, z.B. an PKM anpassen.

3.3 Versionskontrolle

CVS steht für "Concurrent Versions System" und ist die klassische und außerdem freie Variante eines Version Control System, einer Versionskontrolle.

3.3.1 Was ist Versionskontrolle ?

Versionskontrolle bedeutet folgendes: Ein Dokument, sei es der Quellcode für ein Programm, sei es die Mitschrift eines/einer StudentIn o.ä., entsteht über einen Zeitraum. In diesem durchläuft es viele Stadien, in denen es erweitert, umstrukturiert, gekürzt, allgemein: verändert wird. Manchmal merkt man im Nachhinein, daß eine alte Version dieses Dokuments jetzt nützlich wäre. Hat man beim Editieren dieses Dokuments Versionskontrolle verwendet, läßt sich sehr einfach der alte Zustand wiederherstellen, bzw. der alte mit dem neuen vergleichen und kombinieren. So wird in PKM sichergestellt, daß keine Daten verlorengehen, und es wird verhindert, daß der/die BenutzerIn manuell Versionskontrolle betreibt, z.B. durch das Anlegen mehrerer Dateien.

3.3.2 CVS

In CVS spielt sich Versionskontrolle so ab:

1. Man editiert an einem neuen Dokument.
2. Ist man fertig, übergibt man sein Dokument dem CVS-Server (welcher auch auf einem anderen, per Internet verbundenen Computer laufen kann). Dieser legt es an einem Speicherplatz, den er verwaltet, ab. Der Speicherplatz heißt "Repository" und der Vorgang "Check-In". Man wird dabei aufgefordert, einen kurzen Kommentar einzufügen, in welchem Stadium sich das Dokument gerade befindet, um dies später noch nachvollziehen zu können.
3. Will man an dem Dokument weitereditieren, macht man einen "Check-Out" und erhält eine "Working Copy", also eine Version, die man bearbeiten und verändern kann.
4. Hat man einen gewissen Fortschritt erzieht, macht man wieder ein "Check-In".
5. Ist man an älteren Stadien des Dokuments interessiert, kann man sich alle "Check-Ins" ausgeben lassen und es stehen Möglichkeiten zum Vergleichen zwischen den Versionen bereit.

CVS findet in der Open-Source-Bewegung sehr häufig Anwendung, wenn mehrere ProgrammiererInnen an einem Projekt arbeiten, da CVS diesen Prozess auch über das Internet ermöglicht (siehe 2.4)

4 Strukturieren und Ordnen von Informationen

4.1 Probleme der strukturierten Informationsspeicherung

Wenn die zu verwaltende Informationsmenge ein gewisses Maß übersteigt, dann wird sowohl bei auf Papier gespeicherten Informationen als auch bei den Dateien im Computer eine Struktur notwendig. Bei einer "Zettelwirtschaft" sind dies z.B. Schnellhefter, Ordner o.ä., wo Dokumente nach einem gewissen Kriterium abgelegt werden (z.B. ein Ordner nur für Rechnungen). Wenn man nun eine gewisse Rechnung sucht,

dann kann man die Suche auf diesen Ordner beschränken (falls man alle Rechnungen richtig einsortiert hat...), ohne diese Struktur müsste man theoretisch seine gesamten Zettel durchsuchen! Unter gewissen Umständen gibt es sogar noch "Unterordner" in Form von Trennelementen, z.B. die Rechnungen für die Wohnung, das Auto, den Urlaub etc.

Auch bei der elektronischen Datenverwaltung spielt das Konzept eines "Ordners" und "Unterordners" eine Rolle: Dateien werden in einem Dateisystem gespeichert, und gängige Dateisysteme (z.B. FAT unter DOS, ext2 unter Linux) besitzen einen Verzeichnisbaum, wo Dateien in Verzeichnissen und Unterverzeichnissen abgelegt werden können (diese entsprechen den "Ordnern" bei den Papierdokumenten). Einige Vorteile der elektronischen Struktur sind schon in diesem Stadium erkennbar:

- es kann nahezu beliebig viele Unterordner/-verzeichnisse geben, hierbei gibt es keine prinzipiell-physikalischen Grenzen
- die Struktur kann mit wenig Aufwand angepasst werden, da Befehle zum kopieren, verschieben etc. von ganzen Verzeichnisbäumen existieren und mittels "Wildcards" wie "*.jpg" beispielsweise alle jpg-Bilder auf einmal verschoben werden können (bei einem Photoalbum mit echten Bildern muss man dagegen jedes einzelne Bild von Hand verschieben!)

Betrachten wir nun aber folgendes Beispiel: Angenommen die Daten eines/einer ComputerbenutzerIn liegen im Verzeichnis DATA, darunter existiert ein Verzeichnis VORLESUNGEN (der/die BenutzerIn sei ein/e StudentIn!), in dem in mehreren Unterordnern die Vorlesungsmitschriften nach Fach sortiert abgelegt sind (also z.B. PHYSIK, CHEMIE, MATHE). Zunächst ist eine solche Struktur einer einfachen Liste von Dateien ohne Verzeichnisse natürlich überlegen, allerdings hat sie auch ihre Grenzen: der/die StudentIn könnte beispielsweise auf die Idee kommen, seine/ihre Vorlesungsskripte nach Semester ordnen zu wollen, oder nach DozentIn, oder nach irgendeinem anderen Kriterium - in jedem Fall muss er/sie die Verzeichnisstruktur komplett ändern, um die Dokumente nach einem anderen Kriterium zu sortieren. Man muss sich quasi vorher entscheiden, welche Strukturierung der Dokumente am sinnvollsten ist, und dementsprechend den Verzeichnisbaum anlegen. In diesem speziellen Fall wäre das Neu-Sortieren vielleicht verkraftbar, für eine große Wissensdatenbank erweist sich eine Verzeichnisstruktur aber als zu unflexibel.

4.2 Konzept einer relationalen MySQL-Datenbank

MySQL ist eine kostenlose relationale Datenbank, auf die mittels SQL⁸ zugegriffen werden kann. Eine relationale Datenbank besteht zunächst aus Tabellen, in die die Daten eingetragen sind. Entsprechend dem obigen Beispiel existiert also eine Tabelle, in der die Vorlesungsskripte verzeichnet sind, und zwar mit beliebig vielen, frei definierbaren Feldern (für ein Vorlesungsskript sind das dann eben die Felder Fach, Dozent, Semester etc.). Der Hauptvorteil dieser Datenbank sind die sehr flexiblen Abfragemöglichkeiten: Mann kann mittels SQL sehr komplexe Anfragen realisieren, z.B. die folgende: "Alle Vorlesungsskripte des Faches A, die entweder von Dozent B oder C gehalten wurden, und NICHT aus dem Semester D stammen" (dies wird weiter unten demonstriert: siehe 7.3) Des weiteren kann eine "Dokumentendatenbank" viel mehr Meta-Informationen zu den einzelnen Dokumenten enthalten, nach denen dann auch sortiert bzw. gesucht werden kann und es lassen sich Verknüpfungen zwischen den Datensätzen aus verschiedenen Tabellen erstellen und ausgeben.

5 Ausblick

Bereits heute gibt es auf vielen Uni-Campus Funk-LAN⁹, d.h. StudentInnen mit einem entsprechend ausgerüsteten Notebook können von überall auf dem Campus e-mailen, im Internet surfen, also auf riesige Informationsmengen zugreifen. Diese Möglichkeiten werden zunehmend genutzt. Vermutlich sind wir von der Vorlesung, die damit beginnt, daß der Dozent über Funk-LAN das Script der Vorlesung dem Auditorium auf die Notebooks überträgt, nicht mehr weit entfernt. In Zukunft werden sich noch viel weitergehende Visionen realisieren lassen. Auch im Bereich des "Personal Knowledge Mangement" ergeben sich durch

⁸Structured Query Language - die Standard-Datenbankabfragesprache

⁹drahtloses lokales Netzwerk

den technischen Fortschritt in Zukunft viele neue Möglichkeiten. Die Informationstechnologie hat das Potential, das menschliche Leben grundlegend und tiefgreifend zu verändern.

5.1 Das PKM von übermorgen

Bisher funktionierte und funktioniert Informationsverarbeitung und -speicherung in etwa so: Hat man etwas gelesen, wird das Erarbeitete auf Papier niedergelegt. Von da wird es durch Lernen, d.h. mehrmaliges Rekapitulieren, in das Langzeitgedächtnis, zumindest für einen gewissen Zeitraum, übertragen. Mit der Zeit vergißt man jedoch immer mehr davon und muß es sich dann wieder neu erarbeiten und lernen. Nur Dinge, mit denen man mehr oder minder ständig über größere Zeiträume konfrontiert ist, festigen sich bei manchen Menschen für immer. Man muss sich jedoch rechtzeitig daran erinnern, das vor einiger Zeit gelernte zu wiederholen, um sich teilweise Vergessenes wieder klar zu machen, außerdem muß man seine Notizen wiederfinden.

Um darzustellen, wie die Zukunft aussehen könnte, wird der Vergleich zur Arbeitsweise des Computers gezogen. In ihm arbeiten verschiedene Speichermedien zusammen: Die Festplatte ist das langsamste und größte Speichermedium des Computers. Einmal gespeicherte Daten gehen nicht verloren, wenn man sie nicht explizit löscht. Der Arbeitsspeicher arbeitet wesentlich schneller, kann aber wesentlich weniger Daten fassen. Wenn Daten verarbeitet werden, werden sie zuerst von der Festplatte in den Arbeitsspeicher geschrieben, dort kann sehr schnell damit gearbeitet werden, danach werden sie wieder auf die Festplatte zurückgeschrieben. Es lassen sich Parallelen zwischen der Zusammenarbeit von Festplatte und Arbeitsspeicher einerseits und Lang- und Kurzzeitgedächtnis¹⁰ andererseits erkennen. Ein allgegenwärtiges und in den Alltag völlig nahtlos eingebettetes PKM von morgen könnte die Arbeitsweise des menschlichen Gedächtnisses auf eine neue Ebene heben, sodass das menschliche Hirn insgesamt zum relativen Kurzzeitgedächtnis mutiert und eine neue Qualität von Langzeitgedächtnis hinsichtlich Speicherkapazität und vor allem Datensicherheit (-> vergessen) in Form von PKM entsteht: die ständige Verfügbarkeit sämtlichen Denkens, Lernens, Notierens, Exzerprierens und Wissens, das man je getan, gehabt oder sich angeeignet hat.

Solche Visionen haben offensichtlich nicht den Menschen von heute, der sein Notebook aufklappen muß, um neues Wissen einzugeben oder bereits eingegebenes abzufragen, vor Augen. Damit solche Visionen Wirklichkeit werden können, bedarf es völlig neuer Eingabetechniken. Hier sei als mittelfristiges Beispiel Spracherkennung, als langfristiges "Gedankensteuerung" erwähnt (schon heute können körperlich Behinderte ihren Rollstuhl über Gedanken lenken).

Neuartige Verfahren der "Künstlichen Intelligenz" könnten darüberhinaus auch bei PKM den Menschen unterstützen: vielleicht ist es bald möglich, dass Computer durch selbstständiges Strukturieren, einordnen, wiederfinden etc. von Informationen zu einer ganz neuen Qualität von Werkzeug für den Menschen werden (denn das bleiben sie selbst dann!).

5.2 Das Wissen der Menschheit zugänglich machen - eine Vision für die Zukunft

Doch mit der Informationstechnologie bieten sich nicht nur neue Möglichkeit für ein "Personal Information Management", sondern generell hat diese "Kulturtechnik" (dieser Begriff verdeutlicht die strukturelle Vergleichbarkeit der Folgen der Erfindung des Buchdruckes und des Computers/Internets) das Potential, das gesamte menschliche Wissen zu erfassen und jedem Menschen jederzeit zur Verfügung zu stellen, man könnte sagen, das Internet entwickelt sich zu einer "Human Knowledge Base" (HKB). Wenn ein Mensch sich dann eine Frage stellt, so kann er sich per Knopfdruck (oder Sprachbefehl o.ä.) alle Informationen herausuchen, die irgend jemand dazu besitzt oder besessen hat, und die in der "HKB" verzeichnet sind¹¹. Man könnte sich vorstellen, dass es irgendwann möglich sein wird, konkrete Fragen an die HKB zu stellen. Dadurch hätte man bei Bedarf immer die Möglichkeit, einen "Experten" (nämlich die HKB) auf dem benötigten Gebiet zu konsultieren. Dies würde der Menschheit einen enormen Emanzipationsfortschritt bringen, weil kein Mensch mehr aufgrund eines unnötigen Informationsdefizites daraus resultierende, ihm unangenehme Dinge tun müsste. Außerdem würden gesamtgesellschaftlich viele Ressourcen und Kreativität freigesetzt; dies ist eine nach Meinung der Autoren eine sehr positive Vision für die Zukunft.

¹⁰Das Kurzzeitgedächtnis könnte man auch als flüchtigen Cache-Speicher bezeichnen!

¹¹Das HKB könnte auch alle PKM-Daten enthalten, die Menschen hierfür bereitstellen

Teil II

Praktischer Teil

6 Integration von Vim, L^AT_EX, DocBook, CVS für elektronische Mitschriften

6.1 L^AT_EX-Mappings

6.1.1 Mappings für die Strukturelemente

Damit beim Mitschreiben nicht zu viel Zeit für das Eingeben von Strukturelementen verlorengeht, bieten sich Tastaturmappings als schneller Weg an, die gewünschten L^AT_EX-Befehle zu erzeugen und den Cursor auch gleich richtig zu positionieren, beispielsweise beim Einfügen eines Abschnittes, einer `\section{}`:

```
"    insert section level n
map  ,isl1    o<ESC>kO\section{ }<ESC>i
map  ,isl2    o<ESC>kO\subsection{ }<ESC>i
map  ,isl3    o<ESC>kO\subsubsection{ }<ESC>i
```

6.1.2 xfig für Bilder

Um eine kurze Skizze in den Text einzufügen, wird das externe Programm *xfig* gestartet. Zunächst muss man noch in vim einen sinnvollen Namen für das Bild direkt in den Text eingeben, dann den Cursor auf dieses Wort platzieren, und die Tastenfolge `,is` drücken. Nun öffnet sich das Programm *xfig*, man kann die Skizze erstellen, muss diese abspeichern und in das Format *eps* (Encapsulated Postscript) konvertieren (dies geschieht per einfacher Tastenkombination), da L^AT_EX nur dieses importieren kann. Dann muss *xfig* nur noch beendet werden, das Mapping kümmert sich darum, dass ein korrekter L^AT_EX-Befehl zum Einfügen der eben erstellten Grafik in die Datei mit aufgenommen wird:

```
"    insert sketch
map  ,is      :!xfig <cWORD>.fig<CR><CR>\includegraphics{<ESC>A.eps}<ESC>o
```

6.1.3 LyX für Formeln

Das Eingeben von Formeln geschieht in LyX, welches über einen sehr mächtigen, über Tastatur mit viel Übung sehr schnell bedienbaren, Formeleditor verfügt. Aus *vim* heraus wird, analog zu Zeichnungen, wenn der Cursor über einem Namen für die Formel positioniert ist, mit der Tastenkombination `,if` LyX gestartet und automatisch in den “Mathmode” geschaltet, d.h. der Formeleditor gestartet.

```
map  ,if      :!cp newfile.lyx <cword>.tex<CR>:!lyx <cword>.tex<CR><CR>:!sleep
1<CR><CR>:!echo 'LYXCMD:dafe:math-mode' > ~/.lyxpipe.in<CR><CR>
```

6.2 DocBook-XML

Auch für Docbook existieren vim-Mappings, um schneller die entsprechenden Struktur-Tags einfügen zu können. Mehr unter [10]

6.3 XML und Vokabel-Tags und Mappings

In einem späteren Stadium von PKM ist geplant, auch eigene XML-Tags zu erstellen (siehe 9.2), beispielsweise ein “Vokabel-Tag”. Damit können dann während des Lesens eines fremdsprachlichen Textes neue Vokabeln direkt markiert und in eine persönliche Vokabel-Datenbank übernommen werden. Um eine neue

Vokabel mit Tags zu umschließen (ungefähr so: <voc>neueVokabel</voc>), wenn sich der Cursor darauf befindet, kann folgendes mapping verwendet werden:

```
map ,vc bi<voc><ESC>ea</voc><ESC>
```

Eine neue Vokabel kann mittels perl-Skript in die Vokabel-Datenbank übernommen werden:

```
map ,vl :!/home/joe/pkm/perl/vl_insert.pl <word> %:p<NL>
```

6.4 CVS-Mappings

Auch CVS lässt sich mittels vim-Mappings in PKM einbinden. Im momentanen Stadium des Projektes ist dies allerdings noch nicht verwirklicht. Es sollte jede einzelne Datei, die man bearbeitet, mittels cvs erfasst werden, am besten sollte jeder Speichervorgang in vim automatisch mit einem "CVS-Check-In" verbunden sein!

7 Die PKM-Dokumentendatenbank

Dieses Beispiel für eine PKM-Dokumentendatenbank soll die im theoretischen Teil entwickelten Ideen und Konzepte verdeutlichen und exemplarisch veranschaulichen, deckt aber natürlich bei weitem nicht alle Möglichkeiten ab, die Datenbanken für PKM bieten (für weitergehende Ideen siehe 9)

7.1 Mysql und PHPMyAdmin

Eine Datenbank besteht in MySQL zunächst aus Tabellen, in die die Daten eingetragen sind. Im jetzigen Stand von PKM werden alle Dokumente eines/einer BenutzerIn in einer zentralen Tabelle "root" in der Datenbank "pkm" gespeichert. Dort gibt es folgende Felder:

- **id:** laufende Nummerierung
- **path:** vollständiger Pfad (inklusive Dateiname!) des Dokumentes im CVS-Repository
- **type:** Dateityp (es kann genau die Dateitypen enthalten, die der/die BenutzerIn angelegt hat, z.B. "Mitschriften", "Vorlesungen")
- **id_there:** id des Dokuments in der Dokument-spezifischen-Tabelle (s.u.)

Für jeden Dateityp existiert eine eigene Tabelle, in dem weitere, dateitypspezifische Informationen abgelegt sind (z.B. das Fach bei einer Vorlesungsmitschrift). Hierbei ist jedoch zu beachten: Für ein wirkliches "Personal Knowledge Management" muss sich jeder Mensch die PKM-MySQL-Datenbank entwerfen, die genau zu ihm passt, und dabei beispielsweise festlegen, welche "Dateitypen" es gibt und welche Informationen zu den einzelnen Dateitypen gespeichert werden (d.h. wie die Tabelle des "Dateityps" aussieht). Dies kann relativ komfortabel mit PHPMyAdmin, einem PHP-Frontend zu MySQL, geschehen.

7.2 Eintragen von Dokumenten in die PKM-Datenbank

7.2.1 vim-mapping für PHP

```
map ,di :!konqueror localhost:/pkm/db_insert.php?path=%:p<NL>
```

Mittels des obigen Mappings kann ein Dokument direkt aus vim durch drücken der Tastenfolge **,dp** in die PKM-Dokumenten-Datenbank eingetragen werden. Hierzu muss der Anwender ein Web-Formular mit den entsprechenden Angaben ausfüllen, welches im obigen Beispiel durch den Browser "Konqueror" dargestellt wird. Das Web-Formular wird dynamisch vom lokal installierten Web-Server mit Hilfe von PHP-Skripten erzeugt; die Details hierzu finden sich in den nächsten Abschnitten.

7.2.2 Apache und PHP

Apache ist ein freier Open-Source Web-Server, der aufgrund seiner Leistungsfähigkeit und Robustheit der weitverbreitetste Web-server überhaupt ist. PHP ist eine server-seitige Skriptsprache, die den Web-Server erweitert und dadurch die Möglichkeit bietet, dynamische Webseiten zu erstellen (eine andere Möglichkeit hierfür ist CGI-Perl). Dies wird beispielsweise zum Abfragen von Datenbanken über Netzwerke (Intra- oder Internet) eingesetzt, kann aber auch für lokale Datenbankabfragen verwendet werden. In PKM werden drei PHP-Skripte verwendet, um ein Dokument in die PKM-Datenbank einzutragen: *db_insert.php*, *db_process.php*, *db_process2.php*.

7.2.3 db_insert.php

Um ein Dokument zunächst in die root-Dokumententabelle einzutragen, werden zwei Informationen benötigt: der Pfad+Dateiname des Dokumentes und der Dateityp. Das PHP-Skript "db_insert.php" wird von vim mit dem Parameter "path=%:p" aufgerufen, wodurch im Skript der Pfad+Dateiname des aktuellen Dokumentes in der Variable *\$path* gespeichert ist. Im Webformular wird der/die BenutzerIn dazu aufgefordert, den Typ der Datei einzugeben; diese Information steht dann in der Variable *\$type*. Diese beiden Parameter werden an das Skript *db_process.php* übergeben.

7.2.4 db_process.php

In diesem Skript wird ein Dokument entsprechend der Parameter *path* und *type* in die root-Tabelle eingetragen. Dann wird gemäß des Parameters *type* die zu diesem Dokumenttyp gehörige Tabelle geöffnet, die Felder ausgelesen, und der/die BenutzerIn dazu aufgefordert, die notwendigen dokumentspezifischen Angaben zu machen. Diese Eingaben werden im array *field[]* an das Skript *db_process2.php* übergeben, wobei die Anzahl der Parameter natürlich von der Tabellenstruktur abhängt. Deshalb wird in *field[0]* die Anzahl der Parameter gespeichert und ebenfalls mitübergeben.

7.2.5 db_process2.php

In diesem Skript wird das Dokument gemäß der Parameter aus *db_process.php* in die Dokument-spezifische Tabelle eingetragen. Außerdem wird das Feld *id_there* in der root-Tabelle entsprechend dem erfolgten Eintrag in die Dokumenten-spezifische Tabelle gesetzt. Damit ist das Dokument vollständig in die PKM-Datenbank integriert!

7.2.6 Vim-Mappings für Perl

Zusätzlich existieren auch einfache Perl-Skripte, um ein Dokument in die PKM-Datenbank einzutragen. Dies soll die prinzipielle Sprach- und Plattformunabhängigkeit demonstrieren, die Funktionalität bleibt im Moment aber hinter der PHP-Lösung zurück. Es existieren folgende Mappings:

```
“ database
map ,db :!/home/joe/pkm/perl/db_insert.pl % %:p:h<NL>
```

Hiermit kann ein Dokument in die root-Tabelle eingetragen werden, es besteht allerdings (im Moment) noch keine Möglichkeit, das Dokument in die typ-spezifische Tabelle einzutragen.

7.3 Zugriff auf die Datenbank

Mittels SQL lassen sich komplexe Datenbankabfragen realisieren. Um z.B. "Alle Vorlesungsskripte des Faches A, die entweder von Dozent B oder C gehalten wurden, und NICHT aus dem Semester D stammen" aus der Vorlesungs-Tabelle zu extrahieren, ist die folgende SQL-Query zielführend:

```
“SELECT * FROM vorlesungen WHERE Fach=A AND (Dozent=B OR Dozent=C) AND NOT Semester=D”
```

7.3.1 vim-Mappings für PHP

```
“ database query
map ,dq :!konqueror localhost:/pkm/eq_input.php<NL>
```

Das obige mapping ruft ein PHP-Skript (*dq_input.php*) in einem Web-browser auf. Dort kann der/die BenutzerIn seine/ihre gewünschte SQL-Query eingeben, welche dann im PHP-Skript *dq_process.php* bearbeitet wird. Dort wird die Query ausgeführt und das Ergebnis wieder im Web-browser dargestellt ¹².

7.3.2 vim-Mappings für Perl

Mit dem folgenden mapping kann mit Hilfe eines Perl-Skriptes eine SQL-Query ausgeführt werden:

```
“ execute query
map ,eq :!/home/joe/pkm/perl/eq_mysql.pl<NL>
```

8 Fazit zum jetzigen “Stand der Dinge”

Als vorläufiges Fazit der intensiven Beschäftigung mit den elektronischen Werkzeugen zum Erstellen, Speichern, Einordnen usw. von Dokumenten lässt sich feststellen, dass nach Meinung der Autoren schon heute ein Umstieg auf Computertechnologie zum persönlichen Informations- und Wissensmanagement weitestgehend möglich ist. Besonders durch die freie Konfigurierbarkeit fast aller Open-Source Anwendungen und durch die offene Unix-Philosophie lassen sich aus einzelnen mächtigen Tools komplexe Anwendungen erstellen, die insbesondere das Strukturieren und Wiederfinden von einmal gespeicherten Wissen gegenüber der Papier-Version radikal vereinfachen (besonders durch die Möglichkeiten von Datenbanken!). Einige Defizite sind noch im Bereich des schnellen Mitschreibens z.B. von Formeln erkennbar. Ob hier der Computer in der Praxis mit Papier und Stift mithalten kann wird wohl nur ein praktischer Einsatz der PKM-Tools z.B. bei der Mitschrift einer naturwissenschaftlichen Vorlesung zeigen können.

Hierzu sind gemäß der Open-Source-Philosophie alle Interessierten aufgerufen. PKM lebt von vielen BenutzerInnen, die Verbesserungsvorschläge machen und ihre Ideen einbringen. Insofern ist der “jetzige Stand” bei weitem nicht die Endversion für ein PKM in der Praxis, sondern stellt lediglich den Ausgangspunkt für eine hoffentlich glorreiche Zukunft des Projektes dar.

9 mögliche Erweiterungen - die nächsten Schritte bei PKM

9.1 Zugriffsmöglichkeiten auf nicht-persönliche Daten

Damit ist kurzfristig gemeint, der/dem BenutzerIn eine einfache Möglichkeit zu bieten

- Vokabeln nachzuschlagen, also eine Verknüpfung mit einem elektronischen Wörterbuch
- Definition von Begriffen, die z.B. gerade im Text auftauchen, zu erhalten, also eine Verknüpfung mit einem elektronischen Lexikon

9.2 XML bringt neue Möglichkeiten

XML bietet die Möglichkeit, direkt in einem Dokument Vokabeln als Vokabeln, Zitate als Zitate, Fremdwörter als Fremdwörter usw. mittels selbstdefinierter Tags zu kennzeichnen. Diese werden nicht nur in eine Datenbank eingetragen, sondern es bleibt auch die Information genau dort erhalten, wo sie aufgetaucht ist. Dadurch kann man, wenn man das wünscht, sich z.B. alle Vokabeln in einem Dokument farbig markieren lassen und sie in eine Datenbank eintragen. Eine Vokabel-Datenbank bietet die Möglichkeit, Vokabeln, die in mit Hilfe von PKM gelesenen Texten vorkamen, in regelmäßigen Abständen zu wiederholen oder mit

¹²dort sollte man in Zukunft über Hyperlinks direkt auf die gefundenen Dokumente zugreifen können.

Hilfe "Karteikarten-Prinzip" durch einen personalisierten Vokabel-Trainer das Lernen von Fremdsprachen zu effektivieren. Dazu muss zu jeder Vokabel gespeichert werden, in welchem Text sie neu vorkam. Dann kann man beim Lernen von Vokabeln bei Bedarf nachschauen, in welchem Kontext sie standen und wie sie dort verwendet wurden. Dadurch sind Vokabeln auch leichter zu lernen, da sie einen Praxisbezug haben und man sich sicher sein kann, dass sie in in der realen Welt existierenden fremdsprachlichen Texten auch wirklich verwendet werden.

9.3 Integration von nicht-elektronischen Dokumenten

PKM sollte Möglichkeiten bieten, auf Papier enthaltene Information in die Datenbank zu integrieren, und damit zur Bearbeitung auf PKM-Art zu erschließen. Gemeint ist damit konkret, eine Scan- sowie eine OCR Software einzubinden (OCR steht für Optical Character Recognition und bedeutet das Umsetzen von gescannten Texten in Buchstaben, die der Computer versteht, i.d.R. ASCII).

Außerdem könnte auch eine Art "Papier-Originale-Verzeichnis" erstellt werden, in dem wichtige Papier-Dokumente wie Urkunden, Rechnungen etc., die nicht eingescannt wurden, verzeichnet sind (also z.B. eine Katalogisierung der Rechnungen im Rechnungsordner nach bestimmten Kriterien).

9.4 Anbindung an PIM

PIMs erfreuen sich in letzter Zeit großer Beliebtheit: immer öfter trifft man Menschen, die ihre persönlichen Informationen elektronisch verwalten. PIM steht für Personal Information Management. Daran lehnt sich der Name von PKM offensichtlich an, wobei Wissen (das K in "PKM") deutlich einen anderen Akzent setzt als Information. PIM bedeutet die Verwaltung von Adressen und Terminen und bietet oft sogar die Möglichkeit zu deren Verknüpfung (z.B. werden oft die Geburtstage von Personen in der Adressdatenbank automatisch in die Termine-Datenbank eingetragen). PKM ist zwar hauptsächlich etwas anderes als PIM. Trotzdem läßt sich eine breite Schnittstelle zu PIM finden, z.B. ist jedes Dokument mit dem Datum seiner Entstehung verknüpft. Steht der Termin im Voraus fest, könnte seine Beschreibung als Template (Vorlage) für das dort editierte Dokument dienen. Ein Beispiel: Man trägt in seinen Terminkallender eine Diskussionsrunde ein und notiert auch gleich das Thema in Form einer kurzen Zusammenfassung. Diese braucht man eigentlich nicht nochmal eingeben oder manuell übertragen, wenn man dann in der Diskussion sitzt, sondern es sollte automatisch erfolgen. Eine Verbindung von PKM und PIM bietet viele derartige Interaktions- und Verknüpfungsmöglichkeiten.

Literatur

- [1] Mark Kronsbein, Thomas Weinert, Clemens Gutweiler, Peter Petermann: PHP after work, Düsseldorf, 2001
- [2] Allen G. Taylor: SQL für Dummies, Bonn, 1998
- [3] Randal L. Schwartz: Learning Pearl, USA, 1993
- [4] Michael Kofler: Linux: Installation, Konfiguration, Anwendung, Bonn, 1995
- [5] <http://cvsbook.red-bean.com/>
- [6] <http://www.w3.org/XML/>
- [7] <http://www.vim.org>
- [8] <http://www.php.net>
- [9] <http://www.lyx.org>
- [10] <http://lodestar2.com/software/docbook/vimrc>
- [11] <http://www.gnu.org>