Universität Leipzig

Fakultät für Mathematik und Informatik

Institut für Informatik

Teiresias:
Datenbank-basiertes Online-Wörterbuch Neugriechisch-Deutsch

Diplomarbeit

Leipzig, Februar 2007

vorgelegt von

Helmchen, Christian

geb. am: 25.08.1980

Studiengang Informatik

Abstract

Mehrsprachige Anwendungen finden heute eine immer größere Verbreitung. Und zusehends treffen dabei Sprachen mit vollkommen unterschiedlichen Zeichensätzen aufeinander. Wie können aktuelle Datenbanksysteme und Programmiersprachen damit umgehen? Ist eine vollständige Unterstützung solch verschiedener Sprachen in heutigen datenbank-basierten Anwendungen möglich?

Das werde ich in dieser Arbeit klären und ein praktisches Beispiel einer solchen Anwendung vorstellen, das Online-Wörterbuch Teiresias. Es vereint nicht nur die Sprachen Deutsch und Neugriechisch in sich, sondern es nutzt auch linguistisch interessante Verfahren zur Suche und zur Bewertung der Treffer.

Außerdem greift es auf Data-Warehouse-Techniken zurück um eine möglichst hohe Effizienz zu erzielen. Die schwierige Übernahme des Datenbestandes der Vorgängerversion zeigt dabei auch die Probleme bei der Weiterentwicklung von Software und der Umstellung auf neue Technologien auf.

Inhaltsverzeichnis

1. Einführung

Im Web existieren zahlreiche Programme und Online-Wörterbücher für die Übersetzung ins Englische bzw. aus dem Englischen. Doch wie sieht es mit anderen Sprachen aus? Welche Besonderheiten gilt es zu beachten, wenn Sprachen mit völlig unterschiedlichen Buchstaben und Zeichensätzen gemeinsam in einer datenbank-basierten Anwendung zum Einsatz kommen? Und wie beeinflussen deren spezielle linguistische Eigenheiten die verschiedenen Algorithmen zur Verarbeitung von Texten bzw. Strings? Diese Fragen gilt es zu untersuchen und die Ergebnisse an einem praktischen Beispiel zu demonstrieren. Zu diesem Zweck wurde das datenbank-basierte Online-Wörterbuch „Teiresias“ (http://teiresias.uni-leipzig.de) für die Sprachen Deutsch und Neugriechisch (weiter)entwickelt.

1.1. Das Projekt „Teiresias“

Teiresias[1] (griech. Τειρεσίας) war in der griechischen Mythologie einer der bedeutendsten Propheten oder auch Seher [Wi06]. Und genau wie sein antiker Namensvetter soll auch das Wörterbuch den Menschen Antworten auf ihre Fragen in Form von Übersetzungen liefern. Doch während der mythologische Τειρεσίας oft nur Andeutungen machte, soll mit dem Wörterbuch eine konkrete Lösung für das Problem der mehrsprachigen Texte in datenbank-basierten Anwendungen geschaffen werden.
Das Wörterbuch hält detaillierte Informationen zu einem Eintrag bereit, sowohl inhaltlich als auch grammatikalisch. Dazu bietet sich dem Benutzer eine Reihe von sprachwissenschaftlich

1 oder auch lat. Tiresias

interessanten Suchmöglichkeiten von der einfachen Wortsuche über eine so genannte Stammsuche, bei der zu einem gebeugten Wort die Grundform ermittelt wird, bis hin zur weitaus mächtigeren phonetischen Suche, welche ein Wort an seiner Aussprache erkennt, dabei aber die Nachteile des bekannten Soundex-Algorithmus vermeidet.

1.2. Aufgabenstellung

Ziel dieser Diplomarbeit ist die Weiterentwicklung der bestehenden Implementierung des Wörterbuches (siehe Kapitel 2.1.) sowohl auf technischer als auch auf linguistischer Ebene.
Dazu soll zunächst das Schema überarbeitet und erweitert werden, um noch detailliertere Informationen und erweiterte Suchformen zu einem Eintrag aufnehmen zu können. Weiterhin soll untersucht werden, ob und wie weit die verwendete Software das Unicode-Format unterstützt, und wie dieses sich gegebenenfalls durchgängig einsetzen lässt.
Die Daten aus der Vorgängerversion sollen dabei verlustfrei in das neue Wörterbuch übernommen und an das veränderte Schema angepasst werden. Die Suche soll schließlich um die Kombination aus Stammsuche und phonetischer Suche ergänzt werden, wodurch noch vielfältigere Suchanfragen möglich sind.
Als Schnittstelle zur Eingabe und Bearbeitung der Daten durch verantwortliche Autoren soll ein Programm implementiert werden, welches zusätzlich administrative Funktionen bietet. Den zweiten Teil der Implementierung bildet die Web-basierte Suche für jeden interessierten Benutzer. Beide Bestandteile können auf dieselben grammatikalischen Regeln zugreifen um die Konsistenz der Daten und Suchanfragen zu gewährleisten. Außerdem ermöglichen diese Regeln unter anderem die hohe Flexibilität des Wörterbuches.

1.3. Gliederung

Im Anschluss an die Einführung befasst sich Kapitel 2 mit den Voraussetzungen der Diplomarbeit. Nach der Analyse der vorangegangenen Versionen des Wörterbuches werden die konkreten Anforderungen formuliert. Danach erfolgt die Untersuchung der Unicode-

Unterstützung in der vorgesehenen Software, bevor ich einige lexikographische und linguistische Grundlagen näher erläutere und auf die griechische Grammatik eingehe.
In Kapitel 3 geht es dann um die Realisierung des Wörterbuches. Ich stelle zunächst die Architektur und das Datenbankschema vor und erläutere die Konvertierung der bestehenden Datensätze aus dem Vorgänger sowie die damit verbundenen Probleme. Ich stelle die Suchoberfläche vor sowie das Eingabeprogramm und analysiere die bislang im Wörterbuch gesammelten Einträge.
Die verschiedenen Suchstrategien werden in Kapitel 4 genauer beschrieben, ebenso das Verfahren zur Bestimmung der Relevanz eines Treffers. Mit einer Zusammenfassung und einem kurzen Ausblick schließe ich dann in Kapitel 5 die Arbeit ab.

2. Voraussetzungen

In diesem Kapitel werde ich auf die Voraussetzungen des Wörterbuches eingehen. Zunächst werde ich die vorangegangenen Versionen betrachten und anschließend die konkreten Anforderungen darlegen. Als sehr wichtige Voraussetzung werde ich dann die Unicode-Unterstützung der vorgesehenen Software analysieren um mögliche Probleme zu erkennen. Schließlich werde ich die wichtigsten lexikographischen und linguistischen Grundlagen, wie den Aufbau eines Wörterbuches, erläutern und auf einige Besonderheiten der griechischen Grammatik eingehen.

2.1. Die Vorgänger

Es gab vor der in dieser Arbeit vorgestellten Version zwei weitere Versionen des Wörterbuches, auf die ich nun kurz eingehen möchte.

2.1.1. „wbuch1“

Der erste Prototyp des Wörterbuches entstand im Rahmen des Datenbankpraktikums im Jahr 2002/2003. Er basierte auf PHP und MySQL und bot sowohl eine Web-basierte Suche als auch eine Web-basierte Eingabemöglichkeit für die Autoren.
Da der Prototyp auf Grund der damals mangelnden Unterstützung noch nicht auf das Unicode-Format setzte, mussten griechische Zeichen auf andere Weise kodiert werden. Dazu wurde eine spezielle, LaTeX-ähnliche Notation verwendet, bei der jedem griechischen

Buchstaben ein lateinischer Buchstabe zugeordnet wurde. Die speziellen Betonungszeichen Tonos und Dialytika wurden als einfache bzw. doppelte Anführungszeichen jeweils vor dem dazugehörigen Buchstaben notiert (siehe Tabelle 1).

griechisch	***deutsch***	***Kodierung***
καλός	gut	kal'oc
χρώμα	Farbe	xr'wma
ευρωπαϊκός	europäisch	eurwpa“ik'oc

Tabelle 1: Kodierung in „wbuch1“

Für die Darstellung der Ergebnisse im Browser wurden die kodierten Informationen dann wieder in griechische Buchstaben umgewandelt. Daraus ergab sich aber auch ein großer Nachteil: die Eingabe war nur mit einer deutschen (oder auch englischen) Tastatur möglich, griechische Tastaturen wurden nicht unterstützt. Ebenso wenig war es möglich, innerhalb eines Feldes in der Datenbank zwischen den Sprachen zu wechseln, da diese bei der Ausgabe der Ergebnisse sonst nicht zu unterscheiden gewesen wären.
Der Datenbestand für den Testbetrieb des Prototypen umfasste bis zum Jahr 2004 mehr als Eintausend Einträge. Auf diesen war eine normale Wortsuche mit und ohne Platzhalter sowie eine Stammsuche möglich.

2.1.2. „wbuch2“

Die zweite Version des Wörterbuches wurde im Jahr 2004/2005 als Diplomarbeit entwickelt. Sie bestand aus einem Programm zur Eingabe der Daten und - wie der Prototyp - aus einer Web-basierten Suche. Diese war jedoch nicht in PHP realisiert, sondern in Perl über CGI (Common Gateway Interface).
Auch diese Version verwendete noch die LaTeX-ähnliche Notation bei der Eingabe und Speicherung der Daten, sowie die griechische Darstellung im Browser. Das Eingabeprogramm nutzte jedoch bereits die Unicode-Fähigkeiten von Java teilweise aus. Anders als der Prototyp beherrschte die zweite Version auch die phonetische Suche in zwei Stufen, wobei die zweite

nur eingeschränkt funktionsfähig war. Der Datenbestand war zum Ende des Betriebes auf 4281 Einträge angewachsen.

2.2. Anforderungen

Nachdem nun die Vorgänger analysiert wurden, können die konkreten Anforderungen an den Nachfolger formuliert werden. Dazu zählen sowohl technische als auch funktionelle und inhaltliche Aspekte wie verwendete Software, Fragen des Bedienkomforts oder Umfang und Qualität der Daten des Wörterbuches.

Technische Anforderungen

- *Software:* Es soll freie Standardsoftware verwendet werden.
- *Kompatibilität:* Die Implementierung soll plattform- und browserunabhängig sein.
- *Unicode:* Durch die Verwendung von Unicode soll die LaTeX-ähnliche Notation durch „echte“ griechische Buchstaben abgelöst werden.
- *Sicherheit:* Der Schutz vor Angriffen durch illegale Eingabewerte (z.B. MySQL-Injection) muss gewährleistet sein.

Funktionale Anforderungen

- *Suchmechanismen:* Es soll sowohl eine einfache Suche, eine Stammsuche (oder eine vergleichbare Suchmöglichkeit) und eine phonetische Suche in zwei Stufen, sowie die Kombination aus Stammsuche und phonetischer Suche möglich sein.
- *Komfort:* Die Benutzeroberfläche soll intuitiv zu bedienen sein.
- *Eingabehilfe:* Der Autor / Benutzer soll bei der Eingabe griechischer bzw. lateinischer Buchstaben (je nach vorhandener Tastatur) unterstützt werden.
- *Flexibilität:* Die Vorteile eines elektronischen Wörterbuches sollen ausgenutzt werden.
- *Administration:* Verschiedenste grammatikalische Einstellungen sollen von zentraler Stelle aus leicht angepasst werden können.

Inhaltliche und qualitative Anforderungen

- *Vollständigkeit:* Sämtliche griechischen Worte und Begriffe mitsamt deutscher Übersetzung sollen ins Wörterbuch aufgenommen werden können, d.h. die noch zu bestimmenden Eintragstypen müssen alle Worte und Begriffe abdecken.
- *Vorgänger:* Die Daten aus dem Vorgänger sollen verlustfrei übernommen werden. Dabei soll eine ausführliche Fehlerbehandlung und Protokollierung für nachträgliche manuelle Bearbeitung vorgenommen werden.
- *Performanz:* Eine Suchanfrage soll möglichst schnell abgearbeitet werden.
- *Robustheit:* Die Suche soll dabei möglichst fehlertolerant sein.
- *4-Augen-Prinzip:* Es sind 2 Autoren für die Freigabe eines Eintrags nötig. Zumindest einer der beiden soll eine entsprechende Qualifikation haben (Experte).
- *Interaktion:* Der Benutzer soll Kommentare zu Einträgen schreiben, Fragen stellen oder Fehler melden können.

Wie und ob die genannten Forderungen umgesetzt werden können werde ich im weiteren Verlauf der Arbeit klären.

2.3. Unicode-Unterstützung

Ein wichtiger Punkt bei der Umsetzung des Wörterbuches ist die Frage, ob und wie weit die vorgesehene Software das Unicode-Format unterstützt, und ob sich daraus Einschränkungen bei der Implementierung oder bei der Benutzung des Wörterbuches ergeben. Deshalb werde ich in diesem Abschnitt nach einer kurzen Einführung in das Unicode-Format die Unicode-Unterstützung bei MySQL, Java und PHP sowie einigen der populärsten Betriebssysteme und Browser untersuchen. Auf den Webserver Apache werde ich nicht weiter eingehen, da dieser lediglich zur Auslieferung der Ergebnisse der Web-basierten Suche dient.

2.3.1. Das Unicode-Format

Die meisten Zeichensätze sind in der Lage 128 Zeichen (7 Bit, z.B. US-ASCII) oder 256 Zeichen (8 Bit, z.B. ISO-8859-1 / Westeuropäisch) darzustellen. Dies genügt jedoch nicht um sämtliche Zeichen abzudecken, die bei der weltweiten Kommunikation auftreten können. Somit ist die Verwendung mehrerer Zeichensätze erforderlich, was innerhalb eines einzelnen Dokuments schwierig und unpraktikabel ist. Darüber hinaus können zwei Zeichensätze demselben Zeichen unterschiedliche Nummern zuordnen oder dieselbe Nummer für verschiedene Zeichen verwenden. Für den Benutzer äußert sich ein ungeeigneter Zeichensatz dann häufig in unlesbaren Symbolen.

Das Unicode-Format ordnet jedem Zeichen eine einzigartige Nummer zu – unabhängig von Plattform, Programm und verwendeter Sprache (aus [Un06]). Ursprünglich umfasste das Unicode-Format 65.536 Zeichen (16 Bit), als diese jedoch nicht mehr ausreichten wurden weitere 16 dieser so genannten Ebenen hinzugefügt. Die aktuelle[2] Version 5.0 kann daher bis zu 1.114.112 Zeichen ($17 * 2^{16}$) unterscheiden, von denen 99.089 bereits verbindlich belegt sind.

Das gebräuchlichste Unicode-Format im Web ist UTF-8, welches in Browsern sowie auch in Betriebssystemen verwendet wird. Dabei werden Unicode-Zeichen mit variabler Anzahl Bytes kodiert, um möglichst wenig Speicher zu belegen. Für die 128 ASCII-Zeichen genügt schon jeweils 1 Byte (8 Bit, daher die Bezeichnung UTF-8) zur Kodierung, maximal sind 4 Byte für ein Zeichen möglich.

2.3.2. MySQL

Die Open-Source-Datenbank MySQL unterstützt UTF-8 ab Version 4.1 zur Speicherung von Zeichenketten (siehe dazu auch [My06]). Zusätzlich zum Zeichensatz gibt es hier noch verschiedene so genannte *Collations*, die Regeln für den Vergleich und die Sortierung von Zeichen definieren. So werden griechische Buchstaben mit und ohne Betonung nicht unterschieden, wenn man nicht auf binärer Ebene arbeitet.

2 The Unicode Standard, Version 5.0 (Addison-Wesley) erschien am 01.11.2006

Diesen Umstand kann sich das Wörterbuch zunutze machen, sodass der Benutzer bei einer Suchanfrage die Betonung nicht beachten muss. Außerdem werden dadurch bei der Sortierung von Einträgen die Buchstaben mit Betonung, wie in einem Wörterbuch üblich, in die Buchstaben ohne Betonung eingereiht, ähnlich wie Ä, Ö und Ü im Deutschen. Hinzu kommt, dass die Betonung bei manchen deklinierten oder konjugierten Wortformen wechseln – man sagt auch wandern - kann, und die Suche ohne Beachtung der Betonung daher bessere Ergebnisse erzielt. Und schließlich wird durch die Verwendung einer solchen nicht-binären *Collation* die Performanz der Suchanfragen verbessert – wie genau wird in Abschnitt 3.1.2. erläutert.
MySQL ist somit für das Wörterbuch bestens geeignet, zum Einsatz kommt die aktuelle Version MySQL 5.0 Community Server.

2.3.3. Java

Die Programmiersprache Java [Ja06] verwendet intern das Format UTF-16 zur Darstellung von Zeichenketten. Quelldateien im Unicode-Format werden ebenfalls akzeptiert, was die Verwendung von Unicode-Zeichen direkt im Quelltext für diverse Zeichenkettenoperationen ermöglicht.
Somit ist Java geradezu prädestiniert für die gestellte Aufgabe, auch wenn während der Implementierung doch noch ein kleines Problem auftrat. So konnten einige spezielle Zahlzeichen in der verwendeten Java-Version 1.4.2 nicht dargestellt werden, obwohl die interne Verarbeitung scheinbar problemlos funktionierte. Da die betroffenen Zeichen aber sehr selten benötigt werden ist dieser Makel vernachlässigbar.
Zur Kommunikation mit der Datenbank wird außerdem ein JDBC-Treiber (Java Database Connectivity) benötigt. Dazu stellt [My06] den MySQL-Connector/J (verwendete Version: 5.0.3) zur Verfügung. Mit entsprechenden Verbindungsparametern versehen klappt damit die Kommunikation zwischen Java und MySQL im Format UTF-8 problemlos.

```
jdbc:mysql://host/db?user=xyz&password=0815&useUnicode=true&characterEncoding=utf8
```

Listing 2.3.3.: Unicode bei MySQL via JDBC-Treiber

2.3.4. PHP

Die Skriptsprache PHP (PHP: Hypertext Preprocessor, [PHP06]) kann im Gegensatz zu Java keine Unicode-Zeichen direkt im Quelltext interpretieren. Die Verarbeitung solcher Zeichen bzw. Zeichenketten stellt aber dennoch kein Problem dar. Sie können aus der Datenbank gelesen und wieder hineingeschrieben und auch aus Dateien geladen werden, was für die grammatikalischen Regeln wichtig sein wird.
Die Implementierung regulärer Ausdrücke beinhaltet ab PHP 5.1.0 einige spezielle Konstrukte für den Umgang mit Unicode, sodass diese und nachfolgende Versionen den Anforderungen gerecht werden und eingesetzt werden können.
Parameter für die Aktivierung der Kommunikation mit MySQL im Unicode-Format gibt es in PHP noch nicht. Hier müssen die entsprechenden MySQL-Variablen explizit gesetzt werden.

```
mysql_connect("host", "xyz", "0815");
mysql_select_db("db");
mysql_query("SET character_set_client=utf8");
mysql_query("SET character_set_connection=utf8");
mysql_query("SET character_set_results=utf8");
```

Listing 2.3.4.: Unicode bei MySQL via PHP

2.3.5. Betriebssysteme

Auch wenn die Implementierung des Wörterbuches plattformunabhängig ist, habe ich Windows (Windows 2000, Windows XP) und Linux (Debian „Sarge“, SuSE) auf ihre Unicode-Unterstützung untersucht.
Alle Systeme unterstützen offiziell das Unicode-Format, so kann man beispielsweise in Texteditoren einen Unicode-Zeichensatz verwenden und Text über die Zwischenablage kopieren. Ebenso lassen sich bei allen Systemen die Tastenbelegungen umschalten (teilweise nachträglich zu installieren), um so zum Beispiel griechische Buchstaben eingeben zu können.
In einigen Situationen gibt es noch ein paar Stolperfallen, so funktioniert unter Windows die

Eingabe von Unicode-Zeichen auf der Konsole nicht mit der eingestellten Standardschrift, dies bringt jedoch keine weiteren Nachteile mit sich.
Obwohl der Bedienkomfort für den Benutzer damit schon sehr hoch ist, sollte das Wörterbuch eine eigene Unterstützung für die Eingabe in Deutsch und Griechisch anbieten, um nicht auf Funktionen des Betriebssystems angewiesen zu sein.

2.3.6. Browser

Damit die Ergebnisse einer Suchanfrage auch korrekt angezeigt werden können, ist es zwingend notwendig, dass auch die Browser das Unicode-Format beherrschen. Hierzu habe ich einige der verbreitetsten Browser untersucht: Microsoft Internet Explorer, Mozilla / Mozilla Firefox, Opera, Epiphany und Konqueror. Letzterer verwendet dieselbe HTML-Komponente, auf der auch Apple's Safari basiert, welchen ich selbst nicht getestet habe.
Grundsätzlich können alle Browser Webseiten im Format UTF-8 anzeigen. Auch die Verarbeitung von Hyperlinks, welche griechische Buchstaben enthalten, klappt überall gut. Man kann sogar eine URL direkt mit griechischen Buchstaben eingeben, die meisten Browser wandeln diese Zeichen dann in eine Hexadezimalschreibweise um. Einzige Ausnahme bildet hier der Internet Explorer in der getesteten Version 6, er erkennt die manuell eingegebenen griechischen Buchstaben nicht. Auch konnte der Internet Explorer - ähnlich wie Java - einige spezielle Zeichen nicht darstellen. Der Konqueror musste bei der Verarbeitung von Hyperlinks mit griechischen Buchstaben teilweise passen, die Zeichen wurden hier nicht korrekt an das Java-Applet weitergegeben.
Insgesamt ist das Ergebnis jedoch zufriedenstellend, die Anzeige der Ergebnisse ist in fast allen Browsern fehlerfrei möglich.

2.3.7. Fazit

Die Untersuchungen haben eindeutig gezeigt, dass die Unicode-Unterstützung in allen Bereichen, von der Datenbank über die Programmiersprache bis hin zur Darstellung der

Ergebnisse, den gestellten Anforderungen weitestgehend genügen. Einschränkungen gibt es lediglich bei einigen speziellen griechischen Zahlzeichen und der Formatierung der Ausgabe mittels regulärer Ausdrücke. Somit ist die Anforderung der Vollständigkeit nicht ganz erfüllbar, was jedoch auf Grund der Seltenheit dieser Zeichen im Wörterbuch akzeptabel ist.

2.4. Wichtige Grundlagen

In diesem Kapitel werde ich einige linguistische und lexikographische Grundlagen erläutern und auch auf die griechische Grammatik eingehen. Doch zunächst ein paar wichtige Definitionen:

Lexikon / Wörterbuch: „Ein Lexikon ist die Menge der sprachlichen Einheiten, die im aktuellen Verlauf menschlicher Rede bzw. Kommunikation vorkommen. Diese Einheiten werden in der wissenschaftlichen Betrachtung durch unterschiedliche Methoden festgestellt, aus ihrem jeweiligen kontextuellen Zusammenhang herausgelöst und schließlich systematisch in einem Wörterbuch dargestellt.“ (nach [HBL83])

Wortform: Unter einer Wortform versteht man ein gebeugtes (flektiertes) Wort, z.B. *gehe*.

Lexem: Ein Lexem ist der abstrakte Oberbegriff aller Wortformen mit gleicher Bedeutung, z.B. *gehen* als Repräsentant für *gehe*, *gehst*, *geht*, *ging* usw. Die Bezeichnung dieser Einheit des Wortschatzes einer Sprache richtet sich meist nach der Grundform / der Zitierform der repräsentierten Wortformen, z.B. der Infinitiv Präsens Indikativ *gehen* bei Verben. Lexeme sind die Grundbausteine eines Lexikons. Dabei unterscheidet man zwischen einfachen (Simplex) und zusammengesetzten Lexemen (Paralexem).

Lemma: In der Linguistik wird Lemma manchmal als Synonym für Lexem verwendet, in der Lexikographie ist es eine andere Bezeichnung für Grundform / Zitierform.

Phon: Die kleinste Einheit einer sprachlichen Äußerung bezeichnet man als Phon, oder einfacher ausgedrückt: Phone sind die konkreten Laute einer Sprache.

Phonem: Ein Phonem ist die kleinste bedeutungsunterscheidende Einheit einer sprachlichen Äußerung. Es ist aber in der Regel nicht selbst bedeutungstragend. Phoneme sind keine physischen Laute wie die Phone, sondern von ihnen abstrahierte Einheiten. Ein abstraktes Phonem wird durch ein konkretes Phon realisiert, also ausgesprochen bzw. hörbar gemacht.

Minimalpaaranalyse: Die Phoneme einer Sprache lassen sich aus den Phonen ermitteln, denn jedes Phon lässt sich einem Phonem zuordnen. Dazu ersetzt man einfach in einem Wort ein Phon durch ein anderes, z.B. *Daten* - *Raten*. Falls sich die Bedeutung des Wortes ändert (oder ganz verloren geht) kann man die beiden Phone jeweils unterschiedlichen Phonemen zuordnen, andernfalls demselben Phonem. Diese so genannte Minimalpaaranalyse kann man systematisch für alle Phone mit geeigneten Wortbeispielen durchführen und erhält so sämtliche Phoneme einer Sprache.

Allophon: Die Phone, welche demselben Phonem zugeordnet werden können, werden auch als Allophone oder Phonemvarianten bezeichnet, z.B. werden das „ch“ wie in *ich* [ç] und wie in *ach* [x] beide dem Phonem /x/ zugeordnet.

Graphem: Unter einem Graphem versteht man ein Schriftzeichen, die kleinste funktionale Einheit des Schreibsystems einer Schriftsprache.

Morphem: Ein Morphem ist die kleinste bedeutungstragende Einheit einer Sprache. Im einfachsten Fall stellt ein Morphem ein Wort dar, es kann sich aber auch um einen Präfix oder Suffix handeln, z.B. *-keit*. Ein Morphem wird lautsprachlich durch eine Folge von Phonemen und schriftsprachlich durch eine Folge von Graphemen gebildet.

Diathese: In der Linguistik ist die Diathese eine morphologische Kategorie, die das Verhältnis der teilnehmenden Worte beschreibt. Im Deutschen gibt es die beiden Varianten Aktiv und

Passiv, die das Verhältnis zwischen dem Verb und dem Subjekt und / oder Objekt des Satzes beschreiben. Im Griechischen gibt es eine Zwischenform: ein grammatikalisch passives Verb mit aktiver Bedeutung heißt Deponens.

„Roots“: Nach dem Abtrennen der Endung einer Wortform bleibt ein Rest übrig. Dies ist nicht immer der tatsächliche Wortstamm, sondern kann zusätzlich noch andere Morpheme (z.B. eine Vorsilbe) enthalten. Der Einfachheit halber nenne ich diese Formen aber „Roots“.

Die verschiedenen Wortformen werden bei der Stammsuche eine wichtige Rolle spielen, während Phoneme und Grapheme vor allem für die phonetische Suche von Bedeutung sind (mehr dazu in Kapitel 4.).

2.4.1. Aufbau eines Wörterbuches

Wie bereits erwähnt besteht ein Wörterbuch aus Lexemen bzw. Lemmata. Weitere Bestandteile sind ein Abkürzungsverzeichnis, Hinweise zum Aufbau der Einträge sowie zur Grammatik der behandelten Sprachen und eventuell weitere Sachinformationen, die z.B. in einem elektronischen Wörterbuch über entsprechende Hilfefunktionen aufrufbar sind. Das Ganze bildet die Makrostruktur des Wörterbuches, während die Informationen zu einem Eintrag als Mikrostruktur bezeichnet werden (ein Beispiel zeigt Listing 2.4.1.).
Nach [HBL83] versteht man unter der Makrostruktur auch:

> „... die Gesamtheit des Systemgefüges [...] in welchem die einzelnen Lexeme Systemelemente darstellen, zwischen denen Beziehungen (Relationen) bestehen ...“.

Zur Mikrostruktur eines Eintrags nach [Sch87] gehören:

(a) das Lexem in seiner Grundform,
(b) Angaben zur Orthographie (Rechtschreibung), z.B. Silbentrennung durch „|“,
(c) Angaben zur Phonetik (Aussprache, Betonung), üblicherweise in Lautschrift,
(d) weitere orthographische oder phonetische Varianten, z.B. Kurzformen,
(e) Angaben zur Grammatik (Morphologie, Syntax),

(f) diasystematische[3] Angaben, z.B. Sprachstil (Dialekt) oder zeitlich / örtlich begrenzte Verwendung,

(g) Angaben zur Bedeutung, z.B. Erläuterungen, Synonyme und Antonyme,

(h) Belege (Zitate) und evtl.

(i) Verweise auf andere Einträge.

Computer [kɔmˈpjuːtɐ] *m* ηλεκτρονικός υπολογιστής, κομπιούτερ ‹0› *n*

Listing 2.4.1.: Beispiel für einen Wörterbucheintrag

Die meisten Angaben wurden bereits in Teiresias integriert, einige fehlen noch und sollen später hinzukommen (z.B. Lautschrift und Silbentrennung). Die diasystematischen Angaben, von denen es auch eine Reihe von Untergruppen gibt, sind vorerst nicht explizit vorgesehen, jedoch stehen Felder für allgemeine Hinweise zur Verfügung. Verweise auf andere Einträge sind hingegen viel besser umsetzbar als in einem gedruckten Wörterbuch (siehe nächster Abschnitt) und spielen daher auch eine größere Rolle.

Wörterbücher lassen sich nach den verschiedensten Kriterien typisieren (siehe auch [Vo99]):

- Anzahl der Sprachen (einsprachig oder mehrsprachig, unidirektional oder bidirektional),
- Zielgruppe (Schüler, Wissenschaftler oder sogar „Urlauber“),
- Umfang und (Fach)Bereich (Jura, Medizin, Technik, Fremdwörterbuch),
- Detailgrad (Phonetik, Dialekte, Synonyme) oder
- Zugriffsart (alphabetisch, rückläufig alphabetisch, Wortlänge, Häufigkeit).

Teiresias ist ein zweisprachiges, bidirektionales Wörterbuch für Jedermann. Es richtet sich vorrangig an den deutschsprachigen Nutzer und liefert daher detaillierte Informationen zu den griechischen Einträgen, inklusive Synonyme und Phonetik (in Form einer phonetischen Suche, die in Kapitel 4.3. beschrieben wird). Zu deutschen Einträgen werden dagegen nur wenige Details angegeben. Sein Umfang ist noch recht gering, jedoch dauert das Erstellen

3 Ein Diasystem ist ein System aus zwei oder mehr größtenteils sehr ähnlichen Sprachen (z.B. Dialekte). Oftmals handelt es sich aber auch um eine politisch motivierte Trennung der Sprachen (z.B. Rumänisch - Moldawisch).

eines Wörterbuches meist auch mehrere Jahre. Dabei wird kein bestimmtes Fachgebiet abgedeckt, sondern möglichst alle Bereiche. Die Zugriffsart ist auch nicht nur auf eine Variante beschränkt, die technischen Möglichkeiten erlauben diverse Sortierungen und Auswahlkriterien.

2.4.2. Vergleich: gedrucktes und elektronisches Wörterbuch

Die Vorteile eines gedruckten Wörterbuches liegen sprichwörtlich auf der Hand, denn man kann es anfassen - als ein realer Gegenstand ist es vielen Menschen vertrauter als Informationen in elektronischer Form. Außerdem kann man ein gedrucktes Wörterbuch überall hin mitnehmen, ohne an einen Stromanschluss gebunden zu sein. Ein solches Wörterbuch ist in der Regel alphabetisch sortiert und man kann die Sortierung nicht ändern. Sie ist genauso statisch wie alle Informationen zu den Einträgen, was offenkundig der größte Nachteil eines gedruckten Wörterbuches ist.
Der größte Vorteil eines elektronischen Wörterbuches ist dementsprechend seine Dynamik und Flexibilität. Durch verschiedene Suchanfragen lassen sich gezielt einzelne Einträge anzeigen und je nach Implementierung unterschiedlich sortieren und auch formatieren. Gegebenenfalls lassen sich Lesezeichen für einzelne Einträge erstellen, um sie leichter wiederzufinden. Verschiedenste Mechanismen und Web-Technologien[4] erlauben außerdem die komfortable Navigation zwischen Einträgen und verbessern zusehends die Bedienbarkeit. Über Kommentar- oder Editierfunktionen können sich die Benutzer sogar selbst am Wörterbuch beteiligen.
Unterschiede gibt es auch in der Art und Weise der Erstellung eines Wörterbuches. So wird die Planungsphase beim elektronischen gegenüber einem gedruckten Wörterbuch noch um technische Aspekte der Implementierung erweitert. Danach wechseln sich beim gedruckten Wörterbuch Erarbeitungsphase (Einträge erstellen und korrigieren) und Produktion der jeweiligen Auflage ab, wobei die erste Erarbeitung am aufwendigsten ist. Man kann jedoch auch jedes mal von einem neuen Wörterbuch sprechen, sodass nur eine Erarbeitungsphase und

4 z.B. Hyperlinks, AJAX (Asynchronous JavaScript and XML: Konzept der asynchronen Datenübertragung zwischen Webserver und Webbrowser, ermöglicht Anfragen ohne das komplette Neuladen einer Webseite)

Produktion stattfindet. Bei einem elektronischen Wörterbuch entfällt die Produktion an sich, jedoch müssen die Einträge gegebenenfalls für den Zugriff aufbereitet werden. Die Erarbeitungsphase kann dagegen ständig fortgesetzt werden, auch durch die Mitarbeit der Benutzer.

2.4.3. Die griechische Grammatik

Die griechische Grammatik ist weitestgehend durch Regeln bestimmt (siehe dazu auch [Ru02] und [PO05]), welche tief in der Sprachgeschichte verwurzelt sind. Die Kenntnis dieser Regeln sowie einiger ausgewählter Wortformen, die sich natürlich von Wortart zu Wortart unterscheiden, ermöglicht die Bildung weiterer Wortformen und die Bestimmung anderer grammatikalischer Eigenschaften eines Wortes bzw. Eintrags. Es genügt also die Angabe besagter ausgewählter Wortformen, falls sie existieren, sowie einiger zusätzlicher Informationen zur Grammatik, insbesondere bei Abweichungen von den üblichen Regeln, um ein Wort bzw. einen Eintrag sehr genau zu beschreiben.

Wortarten

Das Wörterbuch orientiert sich an der Zehn-Wortarten-Lehre, welche folgende Wortarten kennt: Substantiv, Verb, Adjektiv, Adverb, Pronomen, Präposition, Konjunktion, Numeral, Artikel und Interjektion. Leicht abweichend wurde zusätzlich das Partizip als eigenständige Wortart eingeführt. Diese Klassifizierung eignet sich besonders, da sie aus der antiken griechischen und lateinischen Grammatiklehre[5] stammt.

Für die technische Umsetzung wurde zusätzlich die Fünf-Wortarten-Lehre nach Hans Glinz [Gl67] herangezogen (Abbildung 1), welche alle Hauptwortarten hinsichtlich ihrer Morphologie klassifiziert und damit die zu speichernden Informationen mitbestimmt. Es gibt aber auch Worte, die nur schwer einer bestimmten Wortart zuzuordnen sind: Mehrwortausdrücke (z.B. *Bescheid sagen*), fremdsprachige Ausdrücke, Abkürzungen, Zahlausdrücke (z.B. *14-tägig*) oder kontrahierte Wortformen (z.B. *ins* statt *in das*).

5 Die erste griech. Grammatik verfasste Dionysios Thrax vor mehr als 2100 Jahren, sie kannte 8 Wortarten.

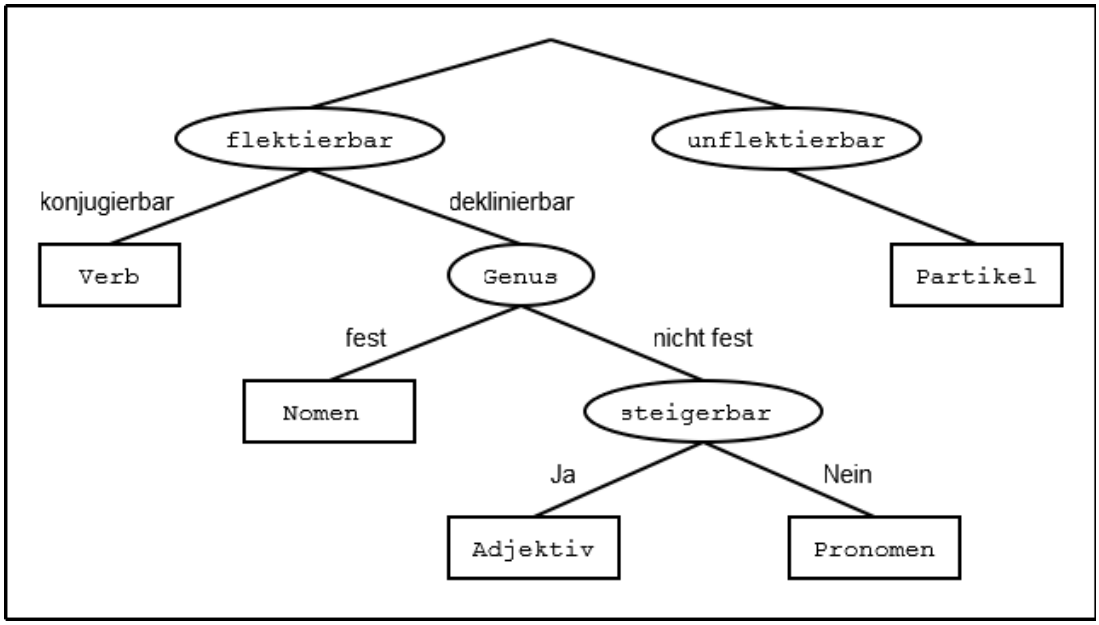

Abbildung 1: Fünf-Wortarten-Lehre nach Hans Glinz

Stoppworte

Eine ganz besondere Gruppe von Worten bilden die so genannten Stoppworte. Dabei handelt es sich um besonders häufig verwendete Worte aus verschiedenen Wortarten. Bei einer Suche nach einem solchen Wort erhält man daher sehr viele Treffer, die größtenteils keine oder niedrige Relevanz haben.

Beispielsweise liefert die Suche nach dem Artikel *der* theoretisch alle maskulinen Substantive sowie eine Vielzahl von Wortgruppen, obwohl für den eigentlichen Artikel nur ein Eintrag notwendig wäre. Um diese Trefferflut einzudämmen werden Worte, die als Stoppwort eingestuft wurden, bei der Suche zunächst ignoriert. Um jedoch den einen passenden Eintrag trotzdem zu finden muss auch eine explizite Umgehung der Stoppworte durch den Autor möglich sein (siehe Abschnitt 3.1.1.).

Kandidaten für Stoppworte sind hauptsächlich Artikel, Präpositionen, Konjunktionen und andere meist kurze Worte, in Kapitel 3.3. findet sich eine kleine Auswahl.

Griechische Verben

Die wichtigste Form oder auch Grundform bei den Verben ist die 1. Person Singular Präsens Aktiv (falls existent, sonst Passiv) Indikativ. Es gibt drei Hauptkonjugationstypen.

(1) Alle stammbetonten Verben werden Typ 1 zugeordnet.
(2) Alle endbetonten Verben gehören zu Typ 2. Letztere werden nochmals nach ihrer 2. Person Singular Präsens Aktiv Indikativ unterteilt:
    (a) Diese kann auf -*άς* enden (Typ 2a).
    (b) Oder sie endet auf -*είς* (Typ 2b).

In [Io92] werden die Verben zusätzlich nach insgesamt 235 Subtypen klassifiziert, welche der Vollständigkeit halber ebenfalls im Wörterbuch angegeben werden können. Weitere Wortformen der Verben (falls sie existieren) sind:

- der aoristische[6] Konjunktiv Präsens Aktiv,
- der Aorist im Aktiv und im Passiv sowie
- das Partizip Perfekt Passiv.

Aus diesen Wortformen lassen sich alle weiteren Wortformen ableiten, Ausnahmen oder Besonderheiten können als Hinweis angegeben werden. Außerdem von Interesse ist die Transitivität (transitiv oder intransitiv) eines Verbs und die Diathese (ist das Verb ein Deponens oder nicht).

Griechische Substantive

Die Grundform der Substantive ist der Nominativ Singular oder Plural, falls der Singular nicht existiert. Substantive haben ein grammatisches Geschlecht (Genus), sie können männlich (Maskulinum), weiblich (Femininum) oder sächlich (Neutrum) sein und sogar männlich und weiblich gleichermaßen (z.B. Berufsbezeichnungen). Außerdem können Substantive nur im Singular (Singularetantum, z.B. *Milch*), nur im Plural (Pluraletantum, z.B. *Ferien*) oder in beiden Numeri auftreten. Der Deklinationstyp eines Substantivs nach [Tr99] wird ebenfalls angegeben und vervollständigt schließlich die wichtigen Angaben, unregelmäßige Substantive müssen allerdings gesondert behandelt werden.

6 Der Aorist beschreibt eine einmalige / abgeschlossene Handlung.

Griechische Adjektive

Bei den Adjektiven gibt es einerseits Vertreter, die männliche, weibliche und sächliche Wortformen aufweisen, und andererseits auch solche, die kein Neutrum bilden (z.B. *ἔγγαμος – verheiratet*). Beim Numerus gilt das Gleiche wie bei den Substantiven, d.h. Adjektive können nur im Singular, nur im Plural oder in beiden Formen auftreten. Adjektive besitzen ebenfalls einen Deklinationstyp, der nach [Tr99] bestimmt wird, und auch unregelmäßige Adjektive müssen gesondert behandelt werden.

Unregelmäßige und sonstige Worte bzw. Wortgruppen

Bei unregelmäßigen Substantiven und Adjektiven ist die Angabe aller Wortformen erforderlich. Diese umfassen die vier Fälle Nominativ, Genitiv, Akkusativ und Vokativ (der Dativ existiert nicht im Griechischen) im Singular und im Plural (falls existent), insgesamt also acht Wortformen.

Da Wortgruppen meist unterschiedliche Wortarten enthalten, werden hier keine weiteren Angaben zur Grammatik gemacht, es wird lediglich der Typ der Wortgruppe (z.B. Redewendung oder Mehrwortausdruck) erfasst. Bei Bedarf werden die Worte unter ihrer jeweiligen Wortart einzeln aufgenommen und entsprechend mit Informationen versehen.

Aus den gegebenen Informationen über den Aufbau von Wörterbüchern und über die griechische Grammatik muss nun ein geeignetes Modell als Grundlage der Implementierung erarbeitet werden, welches ich im nächsten Kapitel vorstellen werde.

3. Realisierung

In diesem Kapitel werde ich die Architektur und das Datenbankschema des Wörterbuches vorstellen und erläutern, wie die Konvertierung der Daten aus dem Vorgänger durchgeführt wurde. Außerdem stelle ich die Suchoberfläche und das Eingabeprogramm vor und fasse die Ergebnisse einiger statistischer Untersuchungen und Tests zusammen.

3.1. Architektur des Wörterbuches

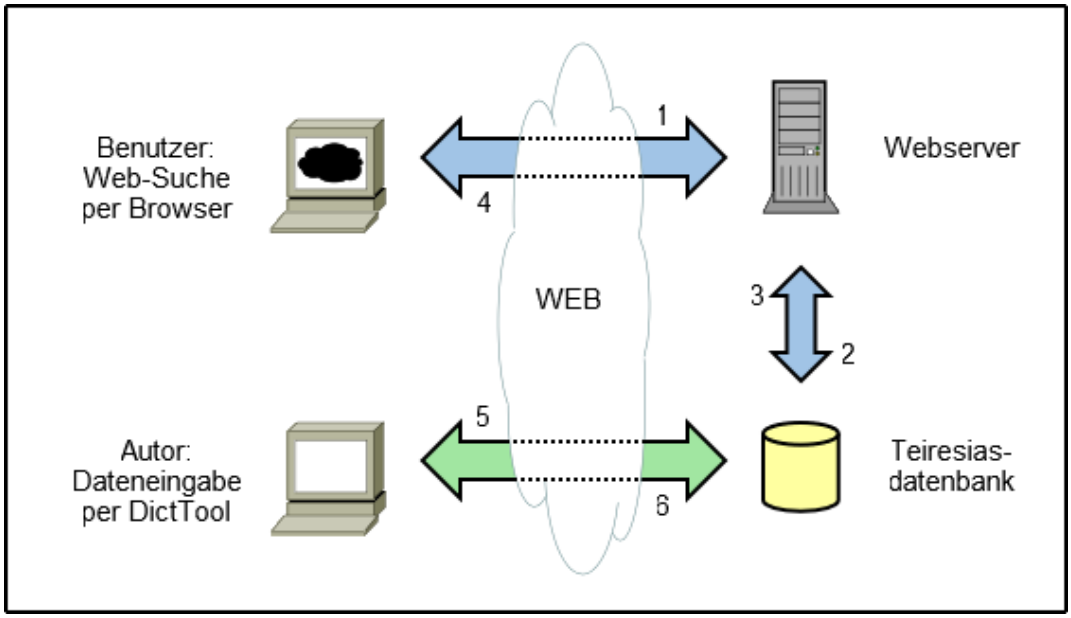

Abbildung 2: Bestandteile des Wörterbuches

1) Anfrage des Benutzers senden
2) bearbeitete Anfrage (SQL) zur Datenbank senden
3) ermittelte Treffer zurückgeben
4) aufbereitete Treffer (HTML) zum Benutzer senden
5) Einträge zur Bearbeitung anfordern
6) bearbeitete oder neue Einträge speichern

Das Wörterbuch besteht neben den eigentlichen Daten aus einer Web-basierten Suchoberfläche und einem Programm zur Eingabe der Daten und zur Administration, dem DictTool (Abbildung 2). Als Datenbank kommt MySQL zum Einsatz. Die Suchoberfläche wurde mit PHP, HTML und einem Java-Applet realisiert und als Webserver wird ein Apache 2 verwendet. Das Eingabeprogramm wurde schließlich in Java implementiert. Ich werde nun zunächst den logischen Entwurf vorstellen und anschließend die Funktionsweise des Eingabeprogramms sowie die Suchoberfläche.

3.1.1. Logischer Entwurf

Die zentrale Rolle im logischen Entwurf übernimmt die Relation *entry*. Sie enthält nicht nur die global eindeutige ID jedes Eintrags, sondern bildet auch die Schnittstelle zu vielen anderen Relationen.

Auf Grund ihrer verschiedenen Flexionseigenschaften (vgl. Kapitel 2.4.) ergeben sich sofort die drei wichtigsten Typen von Einträgen: Verben, regelmäßige Substantive und regelmäßige Adjektive. Auch ein paar andere Worte (z.B. Numerale wie *μερικοί* - *einige*) werden dekliniert wie Adjektive und können daher demselben Eintragstyp zugeordnet werden. Damit auch die unregelmäßigen Vertreter der Substantive bzw. Adjektive ins Wörterbuch aufgenommen werden können, wird ein weiterer Eintragstyp eingeführt: Unregelmäßige Worte. Unregelmäßige Verben gibt es zwar auch (z.B. *τρώω / τρώγω* - *essen* besitzt zwei Grundformen), jedoch lassen sich diese mit wenigen Informationen in den vorhandenen Hinweisfeldern beschreiben (im Beispiel durch einfache Angabe der zweiten Wortform), sodass kein eigener Eintragstyp notwendig ist. Alle übrigen Worte lassen sich dem fünften Typ zuordnen: Einzelworte. Dem gegenüber gibt es schließlich noch einen Typ für Wortgruppen, Redewendungen und andere Begriffe aus mehreren Worten: Ausdrücke. Dazu zählen auch eine Reihe von zusammengesetzten Substantiven, die im Griechischen aus einem Adjektiv und einem Substantiv (z.B. *τραπεζικός υπάλληλος* - *der / die Bankangestellte*) oder einem Substantiv und einem weiteren Substantiv im Genitiv (z.B. *βάση δεδομένων* - *die Datenbank*) gebildet werden können.

Durch eine etwas modifizierte vertikale Partitionierung[7] erhält man damit sechs Relationen für die verschiedenen Typen von Einträgen, welche sämtliche inhaltlichen Angaben enthalten. Dazu gehören Grundform und gebeugte Formen, Angaben zur Grammatik und Hinweise aller Art (z.B. Kontext), sowie für die Suche relevante Informationen (zu sehen in Abbildung 3 am Beispiel der Adjektive). Ein Autor kann außerdem zusätzliche Stichworte angeben um den Eintrag bei einer Suche nach diesen in die Trefferliste mit einzubeziehen - selbst Stoppworte. Die administrativen Daten, wie Status des Eintrags oder letzter Änderungszeitpunkt, werden hingegen vollständig in der Relation *entry* gehalten. Will man nun detaillierte Angaben zu einem Eintrag mit gegebener ID finden, so muss man jedoch nicht alle sechs Eintragstypen durchsuchen. Die Relation *entry* enthält auch ein Attribut *greek_type*, welches den passenden Typ angibt, und somit zusätzliche SQL-Anfragen und Joins erspart, die eine vertikale Partitionierung normalerweise mit sich bringt.

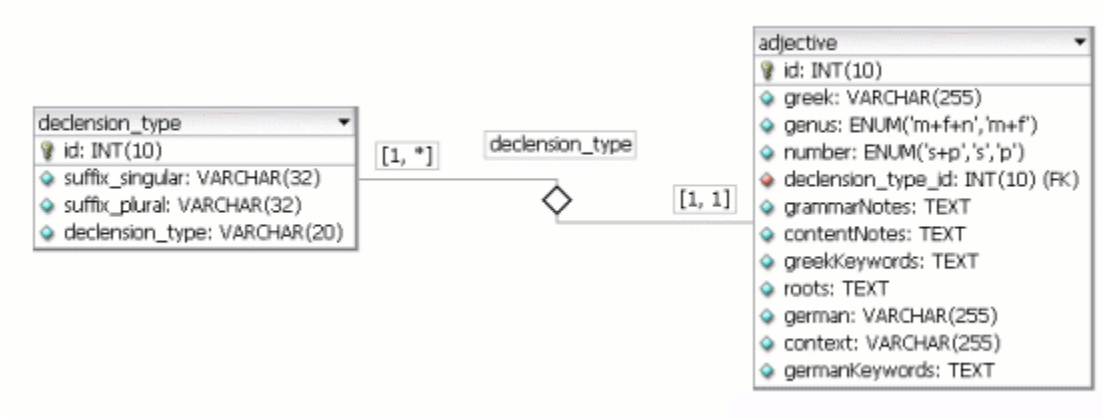

Abbildung 3: Relationen mit den detaillierten Angaben zu Adjektiven sowie deren Deklinationstypen

Eine Assoziation im engeren Sinne besteht zwischen *entry* und den sechs Eintragstypen zwar nicht, aber bereits in [Ra06] wurde die Erweiterung des ER-Modells um Vererbung und Aggregation beschrieben, die auch hier zum Einsatz kommt (Abbildung 4).

Eine Sonderstellung nehmen die Stoppworte ein. Diese werden bei verschiedenen Operationen benötigt und daher separat in der Relation *stopword* gespeichert, um sie effizient abrufen zu können.

7 Im Gegensatz zur üblichen vertikalen Partitionierung wurden einige inhaltliche Angaben in die Unterklassen übernommen, obwohl sie bei allen Eintragstypen gleichermaßen vorhanden sind.

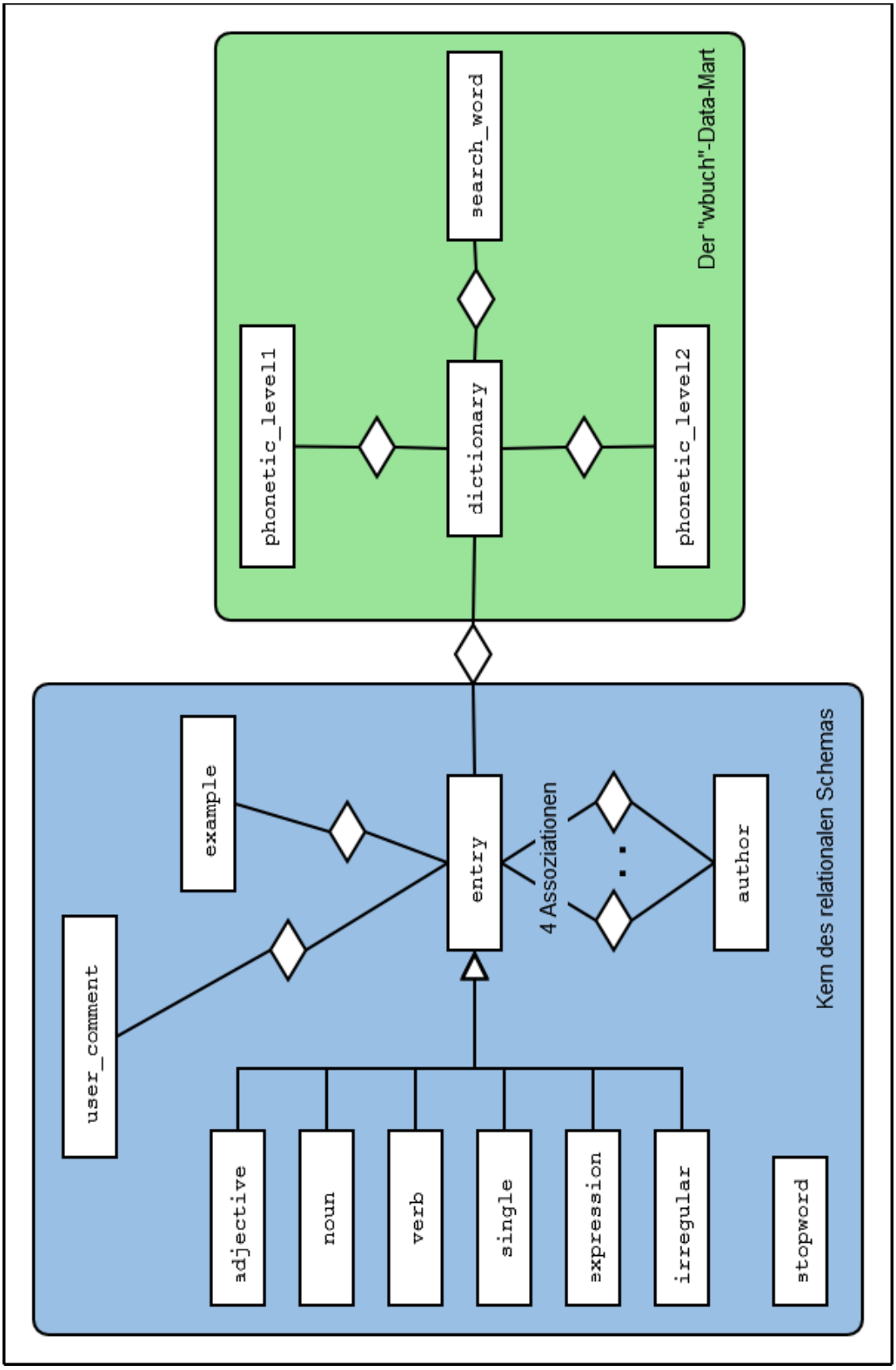

Abbildung 4: Das ER-Modell

Für den Deklinationstyp bei Adjektiven (*declension_type*) sowie den Typ bei Einzelworten (*single_type*), Ausdrücken (*expression_type*) und Unregelmäßigen Worten (*irregular_type*) gibt es zusätzliche Relationen, die eine leichte Änderbarkeit ermöglichen und dem Autor bei der Dateneingabe eine feste, und zugleich überschaubare Auswahl an Werten anbieten (ebenfalls in Abbildung 3 zu sehen, zur besseren Übersicht im ER-Modell in Abbildung 4 ausgeblendet). Eine ähnliche Funktion übernimmt die Relation *context* für alle Eintragstypen, jedoch darf in das entsprechende Feld auch ein beliebiger Wert eingegeben werden.
Das Wörterbuch enthält eine Sammlung von Beispielsätzen, die mit Hilfe der Relation *example* abgebildet werden. Da ein Eintrag auf mehrere Beispiele verweisen kann und ein Beispiel ebenfalls von mehr als einem Eintrag referenziert werden kann, liegt hier eine m:n-Beziehung vor (*entry_has_example*). Für Kommentare, Fehlermeldungen oder Fragen zu einem Eintrag gibt es die Relation *user_comment*, die eine 1:n-Beziehung zu *entry* besitzt. Zur weiteren Kommunikation kann hier neben dem eigentlichen Text noch eine email-Adresse angegeben werden.
In der Relation *author* werden die Autoren des Wörterbuches erfasst. Neben den üblichen Informationen wie Name, Passwort oder email-Adresse hat jeder Autor auch einen Status und gehört einer von vorerst zwei Gruppen an: normale Autoren oder Experten. Zwischen Autoren und Einträgen gibt es insgesamt vier Assoziationen. So übernimmt ein Autor die Rolle des Erstellers eines Eintrags, einer ist der zuletzt bearbeitende Autor und ein Autor der Gruppe Experten prüft einen Eintrag und gibt ihn frei. Wird ein Eintrag gerade bearbeitet so kommt die vierte Assoziation zum Tragen.
Ein weiterer wichtiger Bestandteil des Schemas ist die Relation *dictionary*. Zusammen mit den assoziierten Relationen *search_word*, *phonetic_level1* und *phonetic_level2* bildet sie den so genannten „wbuch“-Data-Mart. Auf dessen Aufbau und Funktionsweise werde ich im folgenden Abschnitt ausführlich eingehen.

3.1.2. Der „wbuch“-Data-Mart

Der „wbuch“-Data-Mart ist eigentlich kein „echter“ Data-Mart, sondern weist auch Merkmale von Data-Warehouses und materialisierten Views auf, je nach Betrachtungsweise.

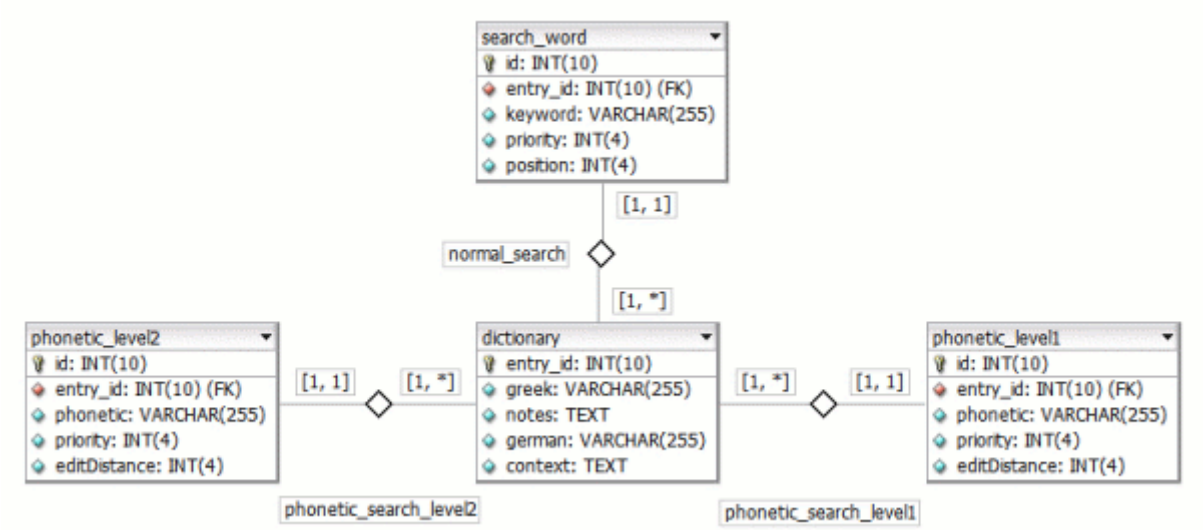

Abbildung 5: Der „wbuch"-Data-Mart

Er besteht aus der Tabelle *dictionary*, die eine kompakte und auf das Wesentliche reduzierte Darstellung aller Einträge enthält, sowie den drei Suchtabellen (lookup tables) *search_word*, *phonetic_level1* und *phonetic_level2* mit zum Teil speziell aufbereiteten Stichworten und dazugehörigen Kenngrößen (Abbildung 4 und Abbildung 5). Zwischen den Einträgen und den Stichworten besteht zwar eine m:n-Beziehung, jedoch sind die Kenngrößen (z.B. Priorität) eintragsspezifisch und daher kommt nur eine 1:n-Beziehung in Frage.

Die Notwendigkeit eines solchen Konstruktes ergab sich aus den speziellen Anforderungen einiger Suchstrategien und der Forderung nach einer performanten Suche. Wie bereits im vorigen Abschnitt erwähnt, sind Suchaufwand und Komplexität der Anfragen durch die vertikale Partitionierung relativ hoch, solange man die Suche nicht auf einen Eintragstyp einschränken kann oder die ID des Eintrags bereits bekannt ist.

Abhilfe kann hier eine spezielle Sicht (View[8]) schaffen, welche einen homogenen Zugriff auf die wichtigsten Informationen aller Einträge bietet. Die Komplexität der Anfragen wird stark reduziert, da nicht mehr sechs Relationen (für die sechs Eintragstypen) mit ihren heterogenen Daten durchsucht werden müssen, sondern nur noch eine Logische. Auch der Suchaufwand wird reduziert, wenn eine materialisierte View verwendet wird. Darüber hinaus wird die Performanz durch die Verwendung geeigneter Suchtabellen und Indizes, welche bereits

8 Eine View ist eine logische Relation, die sich wie eine normale Relation nutzen lässt. Sie wird bei Bedarf berechnet und im Falle einer materialisierten View auch physikalisch zur wiederholten Nutzung gespeichert.

aufbereitete Stichworte und zusätzliche Kennwerte für die verschiedenen Suchstrategien enthalten, weiter stark verbessert. Die Generierung all dieser Daten erfordert jedoch eine Reihe komplexer Operationen auf den Ausgangsdaten der Einträge, die nicht allein vom DBMS durchgeführt werden können. Dazu gehört insbesondere die Vorbereitung der phonetischen Suche durch Transformation aller „gewöhnlichen“ Stichworte nach vorgegebenen Regeln (Näheres dazu in Kapitel 4.3.). Auch eine Verlagerung der Berechnungen weg vom „wbuch“-Data-Mart hin zu den Ausgangsdaten und anschließende redundante Speicherung durch eine materialisierte View ist nicht hundertprozentig möglich. Eine View genügt daher den besonderen Anforderungen noch nicht, sei es eine herkömmliche oder eine materialisierte.

Das Zusammentragen und Aufbereiten der Daten für die Suche erinnert sehr an ein Data-Warehouse, auch wenn die Ausgangsdaten nicht aus verteilten Quellen stammen. Diese Aufgabe übernimmt das DictTool bei der Dateneingabe oder -bearbeitung. Wenn Bedarf besteht, wie etwa nach Änderungen der grammatikalischen Regeln, kann jedoch auch der gesamte „wbuch“-Data-Mart jederzeit neu generiert werden. Die anwendungsspezifische Nutzung ist typisch für einen Data-Mart und führte schließlich zur Namensgebung.

Die kompakte Version eines Eintrags besteht aus der griechischen und deutschen Grundform und den wichtigsten grammatikalischen und inhaltlichen Angaben. Dazu gehören:

- bei den Adjektiven Deklinationstyp und Besonderheiten bei Genus oder Numerus,
- bei den Substantiven Genus und ebenfalls Besonderheiten beim Numerus,
- bei den Unregelmäßigen Worten der Typ und Besonderheiten beim Numerus,
- bei den Verben der einfache Konjugationstyp,
- bei allen anderen Worten die Wortart und
- bei Wortgruppen ebenfalls der Typ.

Durch die Verwendung einer geeigneten *Collation* (siehe Abschnitt 2.3.2.) in der Datenbank werden die betonten griechischen Buchstaben mit ihren unbetonten Pendants gleichgesetzt, wodurch die Menge der verschiedenen Stichworte in den Suchtabellen und damit der Suchaufwand reduziert wird.

Zu jedem Stichwort existiert ein Prioritätswert, welcher bei der Bestimmung der Relevanz eines Treffers verwendet wird. Grundformen haben grundsätzlich eine höhere Priorität als gebeugte Formen, welche wiederum eine höhere Priorität besitzen als vom Autor zusätzlich angegebene Stichworte. Die „Roots“ jedes Eintrags werden ebenfalls in die Liste der Stichworte aufgenommen und mit einer niedrigen Priorität versehen. Über einen geeigneten Schwellwert können verschiedene Suchstrategien damit bestimmte Stichworte ausschließen - die normale Suche filtert so zum Beispiel die „Roots“ heraus.
Die Tabellen *phonetic_level1* und *phonetic_level2* enthalten zu jedem einzelnen griechischen Stichwort aus der Tabelle *search_word* dessen jeweilige phonetische Entsprechung (teilweise auch mehrere Varianten). Der Prioritätswert kann direkt vom Stichwort übernommen werden. Für die phonetische Suche wurde außerdem ein weiteres Maß für die Trefferrelevanz eingeführt: der Edit-Distance-Wert (Details zur Trefferrelevanz in Kapitel 4.5.).

3.1.3. Das Eingabeprogramm: DictTool

Das Programm wurde in Java geschrieben und greift wie in Abbildung 2 erkennbar direkt auf die Datenbank zu. Hierfür wird ein ebenfalls in Java implementierter JDBC-Treiber verwendet.
Beim Login des Autors wird zunächst geprüft zu welcher Gruppe er gehört. Für Experten stehen zusätzliche Funktionen bereit, z.B. das Löschen von Einträgen, Änderung der Stoppworte oder die Generierung des „wbuch“-Data-Mart. Die wichtigsten Funktionen stehen allen Autoren zur Verfügung: das Erstellen und Bearbeiten von Einträgen und Beispielen, die Abarbeitung der Benutzerkommentare und noch einige mehr.

Erstellen von Einträgen

Zum Erstellen eines Eintrags stehen sechs Formularfenster zur Verfügung, für jeden Eintragstyp eines. Hier kann der Autor über Textfelder und Auswahlmenüs sämtliche Informationen eingeben, die zu dem jeweiligen Eintragstyp möglich sind. Das Programm unterstützt ihn dabei durch Integritätsbedingungen. Ein neu erstellter Eintrag hat automatisch den Status *new* und der Autor wird als Urheber und zuletzt bearbeitender Autor eingetragen.

Um Duplikate zu vermeiden wird vor jedem Speichervorgang nach einem Eintrag mit genau demselben griechischen und deutschen Teil sowie Kontext gesucht. Wenn alles in Ordnung ist wird der Eintrag gespeichert und sofort der „wbuch“-Data-Mart aktualisiert, damit der neue Eintrag unmittelbar danach für die Suche zur Verfügung steht.

Suchen von Einträgen

Das DictTool verfügt über eine Reihe von Suchfiltern, die bei der Verwaltung der Einträge hilfreich sind. Dazu gehören:

- Suche nach Stichworten,
- Suche nach Eintragsstatus,
- Suche nach ID,
- Suche nach Bemerkungen oder Benutzerkommentaren,
- Suche nach Beispielen und
- Suche nach gerade bearbeiteten Einträgen.

Die Ergebnisse einer Suchanfrage werden tabellarisch im Hauptfenster des Programms angezeigt und können dort noch einmal nach Eintragstyp gefiltert und spaltenweise sortiert werden. Die technischen Informationen (Eintragstyp, Status, Urheber, letzte Änderung oder Anzahl der Benutzerkommentare) sind per Tooltip verfügbar und auf Knopfdruck kann ein Eintrag zum Bearbeiten geöffnet werden (Abbildung 6).

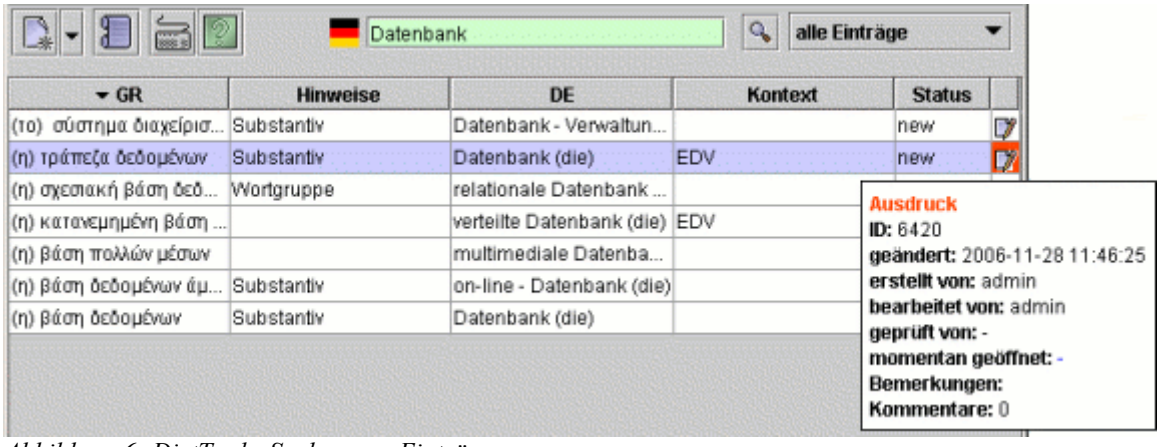

Abbildung 6: DictTool - Suchen von Einträgen

Bearbeiten von Einträgen

Hat der Autor einen Eintrag geöffnet so kann er wie beim Erstellen alle Informationen eingeben bzw. bearbeiten. Zusätzlich kann er jetzt auch Beispiele und Benutzerkommentare zu diesem Eintrag verwalten. Solange ein Eintrag bearbeitet wird steht der bearbeitende Autor im entsprechenden Feld der Datenbank. Kein anderer Autor kann inzwischen diesen Eintrag modifizieren, sondern lediglich ansehen.

Nach jeder Änderung erhält ein Eintrag automatisch den Status *new*. Danach kann nur ein anderer Autor der Gruppe Experten den Eintragsstatus auf *ok* setzen. Ist ein Autor der Meinung ein Eintrag sei fehlerhaft, so kann er dessen Status auch auf *false* setzen. War der Eintrag vorher als *ok* markiert so darf dies nur ein Experte tun. Auch das Löschen ist den Experten vorbehalten, Abbildung 7 zeigt den vollständigen Lebenszyklus eines Eintrags.

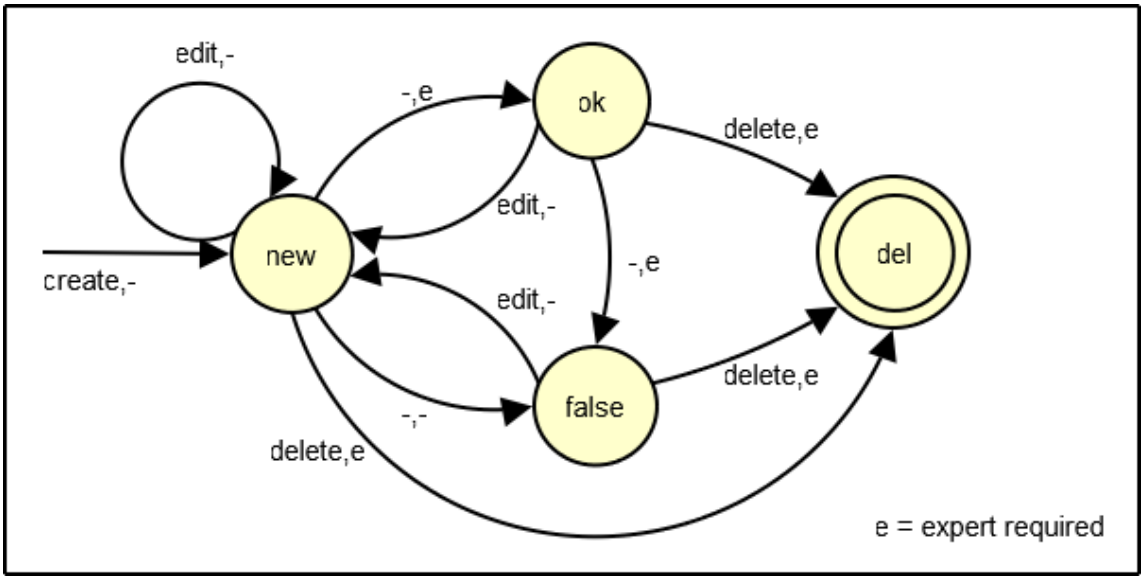

Abbildung 7: Lebenszyklus eines Eintrags

Eingabehilfe

Eine wichtige Aufgabe ist die Unterstützung des Autors bei der Eingabe sowohl griechischer als auch deutscher (lateinischer) Zeichen unabhängig von der physischen Tastatur und den Einstellungen des Betriebssystems.

Daher habe ich für das DictTool eine spezielle Komponente implementiert, welche alle Eingabefelder überwacht und je nach eingestellter Sprache die Eingaben umwandelt

(Abbildung 8). Auch die Eingabefelder wurden erweitert, sodass jedes per Knopfdruck separat von deutsch auf griechisch umgeschaltet werden kann und umgekehrt (es kann aber auch auf eine feste Sprache eingestellt werden). In Abbildung 6 ist ein solches Eingabefeld zu sehen, es ist auf deutsche Eingabe gestellt.

Für eine hohe Performanz bei der Umwandlung werden alle Zeichen mit ihren Pendants in Hashtabellen gehalten, je eine für deutsch-griechisch und für griechisch-deutsch. Somit ist der Zugriff in konstanter Zeit möglich. Zu diesen Zeichen gehören nicht nur Buchstaben, sondern auch Satzzeichen. So sieht das griechische Fragezeichen aus wie das deutsche Semikolon und das griechische Semikolon ist ein im Deutschen nicht verwendetes Zeichen.

Wird nun ein deutsches Zeichen eingegeben, während die Eingabe auf griechisch eingestellt ist, so wird das passende griechische Zeichen aus der Hashtabelle geholt und noch vor dem Einfügen in das Eingabefeld ersetzt. Andersherum funktioniert das natürlich genauso, man kann also mit einer griechischen Tastatur auch deutsche Zeichen eingeben.

Um griechische Buchstaben mit Betonung einzugeben muss man wenigstens zwei Tasten nacheinander drücken. Wird ein solches Betonungszeichen erkannt so gelangt es zunächst in einen Puffer, bis das nachfolgende Zeichen den eigentlichen Buchstaben bestimmt.

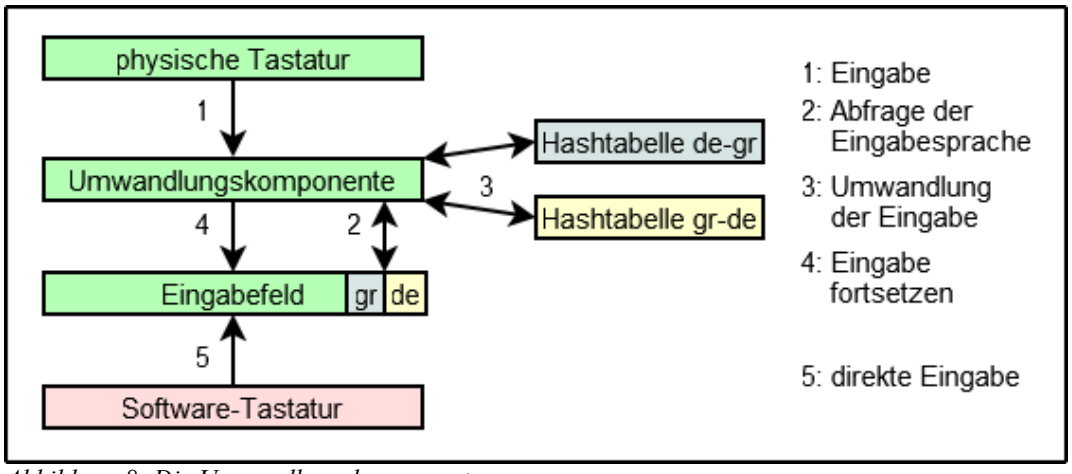

Abbildung 8: Die Umwandlungskomponente

Für die Eingabe per Tastatur bleibt aber dennoch ein Problem bestehen: der Benutzer muss wissen welches deutsche Zeichen welches griechische Zeichen ergibt (oder umgekehrt). Für den geübten Autor stellt das kein Problem dar, für einen ungeübten Benutzer dagegen schon.

Deshalb habe ich als zusätzliche Hilfe eine Software-Tastatur implementiert. Diese ist an die Umwandlungskomponente gekoppelt und schaltet sich automatisch mit den überwachten Eingabefeldern um, sodass der Benutzer z.B. genau sieht welche Taste er für ein *Omega* drücken muss.

Grammatikalische Optionen

Das Feld „Roots“ enthält Angaben für die Stammsuche (siehe Kapitel 4.2.) und kann per Knopfdruck automatisch ausgefüllt werden, indem die anderen relevanten Felder analysiert werden. Der Autor kann jedoch auch manuelle Änderungen vornehmen. Außerdem kann der Autor die Liste der Stoppworte bearbeiten, wenn er zur Gruppe der Experten gehört.

Generierung des „wbuch“-Data-Mart

Nach Änderungen an den grammatikalischen Regeln muss der „wbuch“-Data-Mart neu generiert werden. Ein Autor der Gruppe Experten kann dies mit dem DictTool durchführen, entweder über die grafische Oberfläche oder in einem speziellen Modus auf der Kommandozeile, wenn man z.B. auf einem entfernten Server arbeiten will.

Dazu wird jeder Eintrag aus der Datenbank gelesen und sämtliche enthaltenen Stichworte (einschließlich „Roots“) werden extrahiert. Diese werden mit einem Prioritätswert und einer Referenz auf den Eintrag in die Tabelle *search_word* eingetragen. Anschließend wird jedes einzelne griechische Stichwort in seine phonetische Entsprechung (Stufe 1 und Stufe 2) umgewandelt und in den Tabellen *phonetic_level1* und *phonetic_level2* zusammen mit dem Prioritätswert und einem bei der Umwandlung ermittelten Edit-Distance-Wert gespeichert. Zum Schluss werden noch ausgewählte Eigenschaften des Eintrags untersucht. Wenn diese einen vorher genau definierten Wert besitzen, so werden sie in die kompakte Version des Eintrags für die Tabelle *dictionary* übernommen.

3.1.4. Die Web-basierte Suche

Das Web-Frontend wurde mit PHP, HTML und JavaScript realisiert. Ein Java-Applet sorgt für eine komfortablere Eingabe (vgl. voriger Abschnitt), es geht aber auch ohne Applet und

ohne JavaScript. Die gesamte Web-Oberfläche, einschließlich Applet, lässt sich zwischen Deutsch und Griechisch umschalten und es steht eine ausführliche Hilfe in beiden Sprachen bereit. Um Recherchen etwas zu erleichtern werden die jeweils letzten fünf Suchbegriffe gespeichert und sind jederzeit über das Menü erneut aufrufbar.

Dem Benutzer stehen vier Suchmöglichkeiten zur Verfügung, die ich in Kapitel 4. näher erläutern werde: normale Suche, Stammsuche, phonetische Suche und Kombination aus Stammsuche und phonetischer Suche. Man kann wählen ob man nach einem ganzen Wort suchen will, nach dem Wortanfang oder nach einem Teilwort. Mehrere Stichworte lassen sich per UND bzw. ODER verknüpfen (nur normale Suche) und die phonetische Suche gibt es in zwei Stufen, eine strenge und eine lockere.

Abbildung 9: Anzeige der Ergebnisse in der Web-Oberfläche

Die Ergebnisse einer Suchanfrage werden Zeilenweise angezeigt (Abbildung 9), bei Bedarf auch auf mehrere Seiten verteilt. Per Mausklick lassen sich dann zu jedem Eintrag Details abrufen, entweder im Hintergrund per AJAX und anschließender Einblendung in die aktuelle oder auf herkömmliche Weise über eine neue Seite. Alle Stichworte in den Ergebnissen sind anklickbar, sodass man direkt eine neue Suche starten kann. Um einen Kommentar zu einem Eintrag abzugeben muss man zu dessen Detailansicht wechseln und mit einem weiteren Mausklick öffnet sich das zuständige Formular.

Die Reihenfolge der Ergebnisse hängt von verschiedenen Faktoren ab, auf die ich in Kapitel 4.5. näher eingehen werde. Sonst gleichwertige Treffer werden zum Schluss alphabetisch sortiert, abhängig von der Sprache des Suchbegriffs.

3.2. Konvertierung des Vorgängers

Die Übernahme der Daten aus der vorherigen Version war einer der schwierigsten Schritte bei der Realisierung des Wörterbuches. Gründe hierfür waren einerseits die erhöhte Komplexität des Nachfolgers und andererseits die sehr lückenhafte Dokumentation des Vorgängers. Außerdem wies der vorhandene Datenbestand eine hohe Fehlerquote auf, die es zu bewältigen galt. Wie schon [Le05] feststellte, können solche Fehler meist nur von Domänenexperten aufgelöst werden, jedoch sollten Tools sie dabei unterstützen.

3.2.1. Das Modell

Das Modell des Vorgängers wurde in seinen Grundzügen für die neue Version des Wörterbuches übernommen. Es enthielt ebenfalls sechs Eintragstypen sowie eine zentrale Eintragstabelle, jedoch zusätzlich eine eigene Tabelle für die deutschen Übersetzungen. Es wurden hauptsächlich Felder innerhalb der Tabellen verschoben (die deutschen Übersetzungen befinden sich nun bei den griechischen Daten) oder neue Felder hinzugefügt (z.B. Konjugationstyp bei Verben). Durch die Verwendung von Unicode konnten Felder zusammengelegt werden, die vorher zur Unterscheidung von deutschen und griechischen Inhalten getrennt waren, z.B. Felder für Hinweise zu einem Eintrag. Für eine höhere Flexibilität wurden diverse Hilfstabellen mit vordefinierten Typen (Deklinationstypen, Wortarten) hinzugefügt.
Zur Identifikation von Feldern mit gleicher Bedeutung im alten und neuen Modell bietet sich das so genannte Schema-Matching (mittels einer speziellen Matching-Software wie COMA++, siehe [ADMR05]) an. Dabei wird z.B. per Analyse der Feldnamen ein Mapping zwischen altem und neuem Modell erstellt, welches im einfachsten Fall jedem alten Feld ein neues zuweist. Das Zusammenfassen, Entfernen oder Einfügen von Feldern erschwert die Arbeit eines solchen Matchers allerdings erheblich. Noch schwieriger wird es, wenn sich ganze Tabellen ändern, dann kann ein Domänenexperte diese Arbeit unter Umständen manuell besser verrichten.

Ein Mapping zwischen altem und neuem Modell lässt sich aber auch durch die Analyse der konkreten Daten erstellen. Gleiche Feldinhalte lassen auf Felder mit gleicher Bedeutung schließen. Im Fall des Wörterbuches ergab sich daraus allerdings ein großes Problem: eine erste manuelle Analyse des Modells und der Daten führte zu zwei verschiedenen Mappings, d.h. wenn man von der Korrektheit der Analyse ausgeht wurden Daten im Vorgänger in falsche Felder eingetragen. Durch Vertauschen der beiden betroffenen Felder wurde das Problem schließlich manuell gelöst, eine Matching-Software wäre hier vermutlich (noch) gescheitert.

3.2.2. Die Daten

Die Qualität der Daten hängt maßgeblich vom Prozess ihrer Entstehung ab. Durch Fehler oder Lücken in der Implementierung der Vorgängersoftware wiesen auch die Daten verschiedene Fehler auf, z.B. ungültige Zeichen in der LaTeX-ähnlichen Notation, abgeschnittene oder fehlende Übersetzungen / Wortformen oder auch Duplikate.
Einige Fehlerklassen können auch mittels Data-Mining entdeckt werden, wenn man eine ungefähre Vorstellung hat wonach man suchen muss. Diese Vorstellung wurde oftmals eher zufällig gewonnen, indem einzelne Einträge überprüft wurden. Wurde ein Fehler festgestellt, so wurden anschließend alle anderen Einträge ebenfalls darauf überprüft, um etwaige Regelmäßigkeiten herauszufinden. Solche regelmäßig auftretenden Fehler nennt man systematische Fehler, im Gegensatz zu zufälligen Fehlern, die nur vereinzelt auftreten und nicht reproduzierbar sind.
Bei der Behebung der Fehler helfen Integritätsbedingungen, die mittels Domänenwissen erstellt werden und fehlerhafte Einträge identifizieren. Man spricht auch von einem constraint repair problem. Durch Modifikation dieser Einträge wird deren Integrität wiederhergestellt und sie können übernommen werden. Falls sich Fehler nicht automatisiert beheben lassen, muss diese Aufgabe ein Domänenexperte übernehmen. Hier kann ein Tool zumindest die Art des Fehlers erfassen und in einem Protokoll vermerken oder dem Experten bei der interaktiven Fehlerbehebung assistieren.

3.2.3. Durchführung und Ergebnisse

Zur eigentlichen Konvertierung wurde ein eigenes Programm entwickelt, welches alle alten Einträge umwandelte. Über Integritätsbedingungen wurden zufällige Fehler in den vorhandenen Feldern erkannt und wenn möglich behoben. Die erkannten systematischen Fehler wurden durch entsprechende systematische Korrekturen entfernt. Neue Felder durch das erweiterte Modell wurden mit justierbaren Standardwerten gefüllt oder durch das Parsen vorhandener Hinweisfelder automatisch mit Werten versehen. Abbildung 10 zeigt die Informationen, nach denen die alten Einträge durchsucht wurden, um die Felder Genus und Numerus der Adjektive automatisch zu füllen. Die technischen Angaben wie ID eines Eintrags oder Autor wurden dabei immer komplett neu generiert.

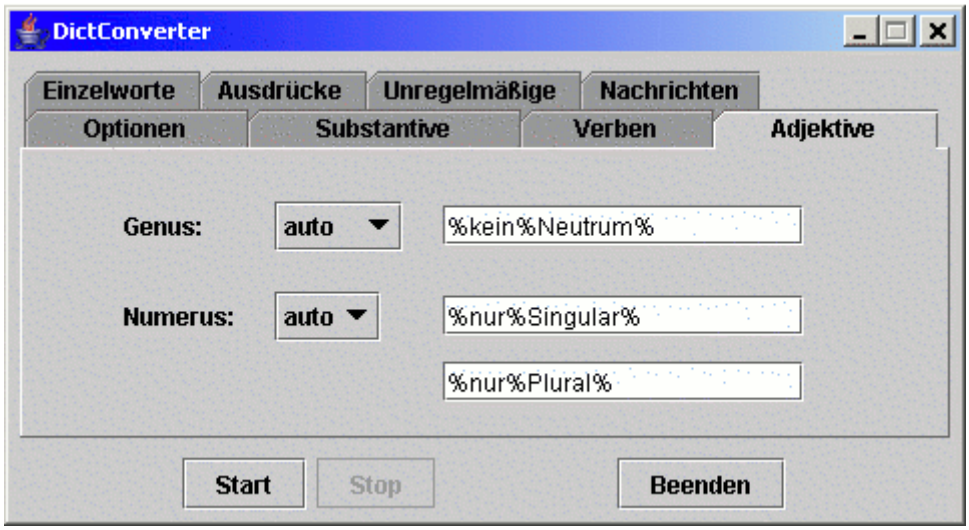

Abbildung 10: Das Konvertierungsprogramm

Sämtliche Fehler, Korrekturen und automatisch ausgefüllten Felder wurden in einem Protokoll gespeichert, genauer gesagt in einer separaten Relation in der Datenbank. Dabei wurden drei Fehlerklassen mit unterschiedlicher Dringlichkeit definiert: Notiz (*note*), Warnung (*warning*) und Fehler (*error*). Später wurden diese als Benutzerkommentare übernommen und konnten so mit dem DictTool von den Autoren (Domänenexperten) abgearbeitet werden.

Es sollte sich herausstellen, dass manche systematischen Fehler nur bei einer begrenzten Teilmenge der Einträge vorkommen, so gab es z.B. etwa 150 Duplikate. Vermutlich wurden bei einem früheren Importvorgang zu Testzwecken diese Einträge doppelt verarbeitet, eine genauere Analyse war leider nicht mehr möglich.

Nach der Konvertierung aller 4281 Einträge des Vorgängers hatten sich 838 Einträge im Protokoll angesammelt, davon 126 Notizen, 422 Warnungen und 290 Fehler. Die tatsächliche Fehlerzahl liegt jedoch etwas höher, da jeder Eintrag nur einmal mit der jeweils höchsten ermittelten Dringlichkeit im Protokoll aufgeführt wurde.

Nach erfolgter Konvertierung wurde das Programm nicht weiter gepflegt, sodass es heute nicht mehr ganz auf dem aktuellsten Stand des Wörterbuches ist, jedoch war es von vornherein nur für den einmaligen Einsatz vorgesehen.

3.3. Das Wörterbuch in Zahlen

Auch wenn die bislang im Wörterbuch enthaltenen Einträge nicht repräsentativ für die gesamte griechische Sprache sind, so kann man doch anhand statistischer Untersuchungen einige Prognosen erstellen, z.B. wieviele Stichworte ein Eintrag eines bestimmten Typs im Schnitt generiert.

Eintragstyp und Anzahl	***# Stichworte***	***# Phonetische Stichworte Stufe 1***	***# Phonetische Stichworte Stufe 2***
Substantive: 3.093	14.136 / 16.209	9.587 / 9.587	10.244 / 10.244
Verben: 1.090	11.041 / 11.177	9.292 / 9.292	10.392 / 10.392
Adjektive: 1.796	6.660 / 6.680	3.769 / 3.772	4.059 / 4.062
Einzelworte: 411	1.385 / 1.423	737 / 745	772 / 780
Ausdrücke: 998	6.697 / 7.693	4.127 / 4.652	4.303 / 4.828
Unregelm.: 24	195 / 197	108 / 109	117 / 118
Gesamt: 7.412	**40.114 / 43.379**	**27.620 / 28.157**	**29.887 / 30.424**

Tabelle 2: Einträge und Stichworte (mit / ohne Filterung der Stoppworte), 05.02.2007

Mit Stand vom 5. Februar 2007 enthält das Wörterbuch 7412 Einträge und 177 zusätzliche Beispielsätze. Deren Verteilung auf die einzelnen Eintragstypen ist in Tabelle 2 zu sehen. Außerdem wurde der „wbuch“-Data-Mart einmal unter Berücksichtigung der zu diesem Zeitpunkt festgelegten Stoppworte (34 Worte) und einmal ohne diese generiert, um festzustellen wie groß die Einsparungen bei der Suche sind.

Wie in Abbildung 11 zu erkennen ist, machen die Stoppworte ca. 7,5% aller Stichworte aus. Bei den Ausdrücken sind es immerhin 13% und bei den Substantiven ebenfalls fast 13%. Dabei handelt es sich im Falle der Substantive bisher ausschließlich um deutsche Stoppworte, was auch die gleich bleibende Anzahl phonetischer Stichworte in Tabelle 2 erklärt. Bei den übrigen Eintragstypen liegt der Anteil der Stoppworte bei nur wenigen Prozent. Die häufigsten Stichworte sind *in* (59 mal) und *sein* (53 mal). Diese wären die nächsten Kandidaten für die Liste der Stoppworte. Die häufigsten Worte überhaupt (inklusive aller Stoppworte) nach den Artikeln *der*, *die*, *das* wären *το* (182 mal) und *sich* (130 mal).

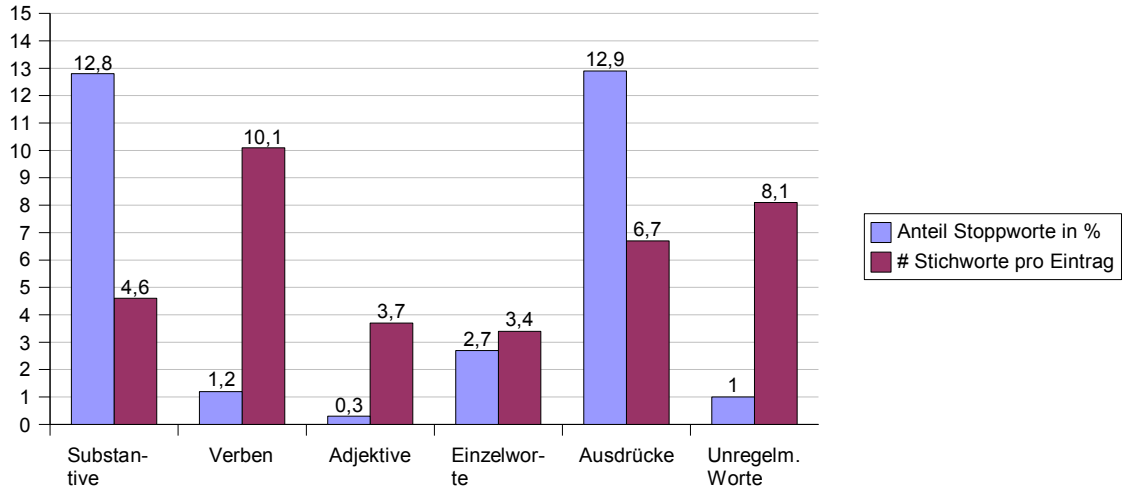

Abbildung 11: Stoppwortanteil und Stichworte pro Eintrag

Ebenfalls aus Abbildung 11 ersichtlich ist die durchschnittliche Anzahl generierter Stichworte pro Eintrag bei den verschiedenen Eintragstypen. Die Verben nehmen dabei die Spitzenposition ein mit durchschnittlich 10 Stichworten pro Eintrag, dicht gefolgt von den Unregelmäßigen Worten. Deren Wert ist jedoch mit Vorsicht zu genießen, da nur sehr wenige Einträge dieses Typs vorhanden sind. Die Werte sind insgesamt nicht sehr überraschend, da bei Verben und Unregelmäßigen Worten die meisten gebeugten Worte angegeben werden, die

somit auch viele Stichworte generieren. Die Ausdrücke folgen mit etwa 7 Stichworten pro Eintrag, danach Substantive, Adjektive und schließlich erwartungsgemäß die Einzelworte mit den wenigsten Stichworten pro Eintrag.

Zum Abschluss habe ich noch die Zeit zum Generieren des „wbuch“-Data-Mart mit dem DictTool in 5 Durchgängen gemessen. Auf einem durchschnittlichen PC (Athlon XP 2400+, 512 Mb RAM) dauerte der komplette Vorgang im Mittel 6:35 min, wobei die Datenbank auf demselben Rechner lief. Bei einem früheren Test über eine Internetverbindung (DSL 2000) hatte die Generierung über eine Stunde gedauert, obwohl es zu diesem Zeitpunkt weniger Einträge gab. Die Ursache dafür ist die hohe Datenmenge, die zwischen Datenbank und DictTool für die komplexen Berechnungen übertragen werden muss. Daher ist die Durchführung auf einem entfernten Rechner nicht zu empfehlen. Falls nur per Kommandozeile auf den Server mit der Datenbank zugegriffen werden kann, so sollte man den speziellen Modus zur Generierung des „wbuch“-Data-Mart per Konsole verwenden.

4. Suchfunktionalität

In diesem Kapitel werde ich auf die verschiedenen Suchstrategien mit ihren Vor- und Nachteilen eingehen sowie auf die jeweilige Bestimmung der Relevanz eines Treffers.

4.1. Einfache Suche

Wie die anderen Suchverfahren verfügt die einfache Suche über drei Optionen für die Genauigkeit der ermittelten Treffer:

- Suche nach ganzem Wort,
- Suche nach Wortanfang (die so genannte Wörterbuchsuche) und
- Suche nach Teilwort.

Die Ergebnismenge steigt dabei immer weiter an, und zwar umso mehr je kürzer der Suchbegriff ist. Nur bei der einfachen Suche ist die Eingabe deutscher Begriffe möglich. Mehrere Suchbegriffe lassen sich wahlweise per UND bzw. ODER verknüpfen, auch deutsche und griechische Begriffe gemischt.
Als Basis dient der einfachen Suche die Menge aller Stichworte im „wbuch“-Data-Mart. Dabei werden alle „Roots“ herausgefiltert, indem der Schwellwert für die Priorität niedriger angesetzt wird als der aller „Roots“. Die einfache Suche berücksichtigt bereits Lautverschiebungen[9] (Tabelle 3) und löst diese auf, d.h. in allen Stichworten sowie im Suchbegriff wird eine einheitliche Schreibung durch Ersetzung einer Variante erzwungen.

9 Durch die Lautverschiebung existieren zum Teil zwei völlig korrekte Varianten eines Wortes.

Der Vorteil der einfachen Suche ist die leichte Handhabung, der Hauptnachteil die fehlenden Möglichkeiten der anderen Suchstrategien.

von	***nach***	***von***	***nach***
πτ	φτ	φθ	φτ
κτ	χτ	χθ	χτ
σχ	σκ	νδ	ντ
σθ	στ		

Tabelle 3: Lautverschiebungen

4.2. Stammsuche

Das Ziel der Stammsuche ist das Auffinden von Einträgen, auch wenn der Suchbegriff in einer anderen Wortform vorliegt als die Stichworte im „wbuch"-Data-Mart, d.h. der Suchbegriff liegt in einer nicht direkt im Wörterbuch enthaltenen gebeugten Form vor. Die Stammsuche arbeitet nur mit griechischen Worten.
Das Ziel wird erreicht indem man sowohl die Stichworte im Wörterbuch als auch den Suchbegriff auf denselben Term / dasselbe Teilwort zurückführt, um so Gemeinsamkeiten zu entdecken. Dieser Vorgang wird auch als Stemming bezeichnet. Im Idealfall ist dieses gemeinsame Teilwort auch tatsächlich der Wortstamm (engl. *stem* - *Stamm*), jedoch können je nach verwendetem Algorithmus noch weitere Affixe (Vor- oder Nachsilben) enthalten sein. Dennoch verwende ich die Bezeichnung Stammsuche in diesem Zusammenhang, da es sehr ähnliche Verfahren sind.

4.2.1. Verfahren der Stammwortreduktion

Das einfachste aber zugleich auch aufwendigste Verfahren ist die Angabe des Stammes zu jedem einzelnen Wort einer Sprache. Damit können auch unregelmäßige Stämme sicher erkannt werden, jedoch ist die Erstellung einer vollständigen Liste sehr mühsam.

Ein anderes Verfahren orientiert sich an der successor variety (Nachfolgervielfalt) eines Wortes bzw. Strings. Diese gibt an wieviele verschiedene Buchstaben nach dem String folgen können, um neue gültige Worte zu bilden. Nun berechnet man die successor variety jedes Anfangsstrings des Wortes. Mit zunehmender Länge dieser Strings nimmt die successor variety zunächst ab, um dann an einer Morphemgrenze plötzlich anzusteigen. Hat man die Morpheme des Wortes damit identifiziert (z.B. durch Trennen an den Stellen mit besonders hoher successor variety) kann man den Stamm daraus ermitteln. Das Problem dabei ist allerdings zu wissen, was gültige Worte sind. Dabei kann man sich nur auf die bereits vorhandenen Einträge stützen.
Die dritte Möglichkeit ist das Entfernen von Affixen nach gewissen Regeln. Dabei wird immer der längste mögliche Affix abgetrennt. Der bekannteste Algorithmus hierbei ist der Porter-Algorithmus für die englische Sprache. Dieser lässt sich jedoch leider nicht für stärker flektierende Sprachen wie Griechisch verwenden.
Bei der Stammwortreduktion kann es zu verschiedenen Fehlern kommen. Werden zwei nicht verwandte Worte auf denselben Stamm reduziert, so spricht man von Overstemming. Das Gegenteil ist der Fall, wenn zwei verwandte Worte auf unterschiedliche Stämme abgebildet werden, man spricht von Understemming.

4.2.2. Die Stammerkennung in Teiresias

Das Wörterbuch verwendet einen Algorithmus zur Abtrennung von Flexionsendungen, jedoch nicht nach gewissen Regeln, sondern mittels Listen von möglichen Endungen. Diese werden in Textdateien tabellarisch gespeichert und sind somit leicht erweiterbar. Sowohl das DictTool als auch die Web-Oberfläche greifen darauf zu.
Somit betrifft die Stammsuche alle flektierbaren Wortarten (vgl. Abschnitt 2.4.3. und Abbildung 1), und zwar hauptsächlich Substantive, Adjektive und Verben.
Der Suchbegriff wird dann auf alle diese Endungen getestet und Passende werden abgeschnitten. So erhält man ein oder mehrere Teilworte des Suchbegriffes, mit dem die eigentliche Suche schließlich durchgeführt wird. Diese Teilworte müssen keine echten Stämme sein, sondern können noch Affixe enthalten.

Die Vorbereitung der Stichworte findet bereits bei der Eingabe der Daten durch die Autoren statt (siehe auch Abschnitt 3.1.3.). Für alle betroffenen Stichworte werden die so genannten „Roots“ generiert und dann in den „wbuch“-Data-Mart aufgenommen, wo sie der Stammsuche zur Verfügung stehen. Die Anzahl der verschiedenen „Roots“ ist von Eintrag zu Eintrag unterschiedlich, so kann ein Verb bis zu vier Stämme haben (je einen für Präsens und Aorist im Aktiv und Passiv).

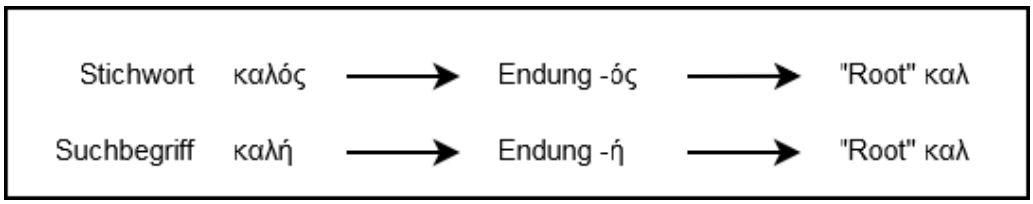

Abbildung 12: Beispiel für zwei verwandte Worte

Der große Vorteil der Stammsuche ist die Möglichkeit verwandte Worte zu finden (Abbildung 12). Damit liefert die Suche auch Treffer wenn der Suchbegriff selbst gar nicht im Wörterbuch enthalten ist. Der Nachteil ist die mühsame Erstellung der Listen mit möglichen Endungen, jedoch genügt die einmalige Durchführung dieser Auflistung.

4.3. Phonetische Suche

Die phonetische Suche ist die wohl mächtigste und zugleich weitestgehend fehlertolerante Suchmethode. Ihr Ziel ist es, griechische Worte nicht nur an der Schreibweise, sondern auch an der Aussprache zu erkennen, da für eine korrekte Schreibung oft tiefere Kenntnisse der Sprachgeschichte notwendig sind, die man nicht immer voraussetzen kann.

So gibt es z.B. für das Phonem /i/ im Griechischen insgesamt 6 Grapheme bzw. Graphemfolgen (ι, η, υ, ει, οι, υι), selbst ohne Betonungszeichen, und auch für andere Phoneme existieren teilweise mehrere Grapheme (Details siehe [Ru02]). Weitere unhörbare Varianten sind Doppelkonsonanten, die sich nicht von den kurz gesprochenen einfachen Konsonanten unterscheiden lassen.

Bei der phonetischen Suche wird der Suchbegriff auf eine Klasse von gleich klingenden Worten abgebildet, diese phonetische Wortklasse ist dann das Ergebnis der Suche.

4.3.1. Der Soundex-Algorithmus

Der bekannteste Algorithmus zur Analyse phonetischer Merkmale eines Wortes ist der Soundex-Algorithmus von Robert C. Russell aus dem Jahre 1918, der für die englische Sprache entwickelt wurde. Dieser bildet ein Wort auf einen vierstelligen Code ab. Dabei ist der erste Buchstabe auch das erste Zeichen des Codes, danach werden alle Konsonanten auf einige wenige Klassen abgebildet und mit einer Ziffer kodiert. Nach maximal vier Stellen bricht die Kodierung ab und der Rest des Wortes entfällt. Auf Soundex basierende Algorithmen wie Metaphone und Phonex arbeiten ähnlich, auch wenn sie modifizierte Regeln verwenden (Details zu den Algorithmen in [Si03]).
Der Soundex-Algorithmus birgt zwei große Probleme in sich: er ist nur bei der Englischen Sprache effektiv einsetzbar und er ist durch die maximal vier Stellen sehr grob. Für das Neugriechische lässt er sich leider nicht zufriedenstellend anwenden. Die Ursache liegt unter anderem in der fehlenden Unterstützung für Grapheme, die mehrere Phoneme repräsentieren (so entspricht „ψ“ der Folge „ps“) und umgekehrt. Auch wird der Kontext eines Zeichens nicht berücksichtigt, was für die griechische Sprache unabdingbar zur Feststellung der Aussprache ist.

4.3.2. Die phonetische Suche in Teiresias

Um eine phonetische Suche im Griechischen zu realisieren benötigt man eine eigene Regelmenge, die wesentlich komplexer als beim Soundex-Algorithmus ausfällt. Von großer Bedeutung für die Aussprache eines Graphems ist dabei dessen Kontext, d.h. es müssen auch die nachfolgenden Grapheme analysiert werden (und manchmal sogar das vorherige). Aus den Ausspracheregeln aus [Ru02] folgt, dass bis zu drei aufeinander folgende Zeichen betrachtet werden müssen.
Auf Basis dieser Regeln wurde ein spezielles zweistufiges Verfahren entwickelt. Dabei steuert ein Regelinterpreter die phonetische Analyse eines Wortes. Dessen Regeln lassen sich beliebig per Texteditor anpassen und erweitern. Eine Übersicht aller Regeln für beide Stufen findet sich in [So07].

Für alle Regeln gilt, dass stets die längste mögliche Zeichenfolge zuerst betrachtet wird, also zuerst die Dreierkombinationen, danach die Zweierkombinationen und zuletzt einzelne Zeichen (Abbildung 13). Sobald eine passende Regel gefunden wurde, wird sie angewendet, d.h. der zur Regel gehörende Code wird an den gerade bearbeiteten String angehängt und je nach Schrittweite der Regel wird die Transformation bei einem der nächsten Zeichen fortgesetzt. Gleichzeitig wird der Edit-Distance-Wert des aktuellen Wortes um den in der Regel angegebenen Wert erhöht.

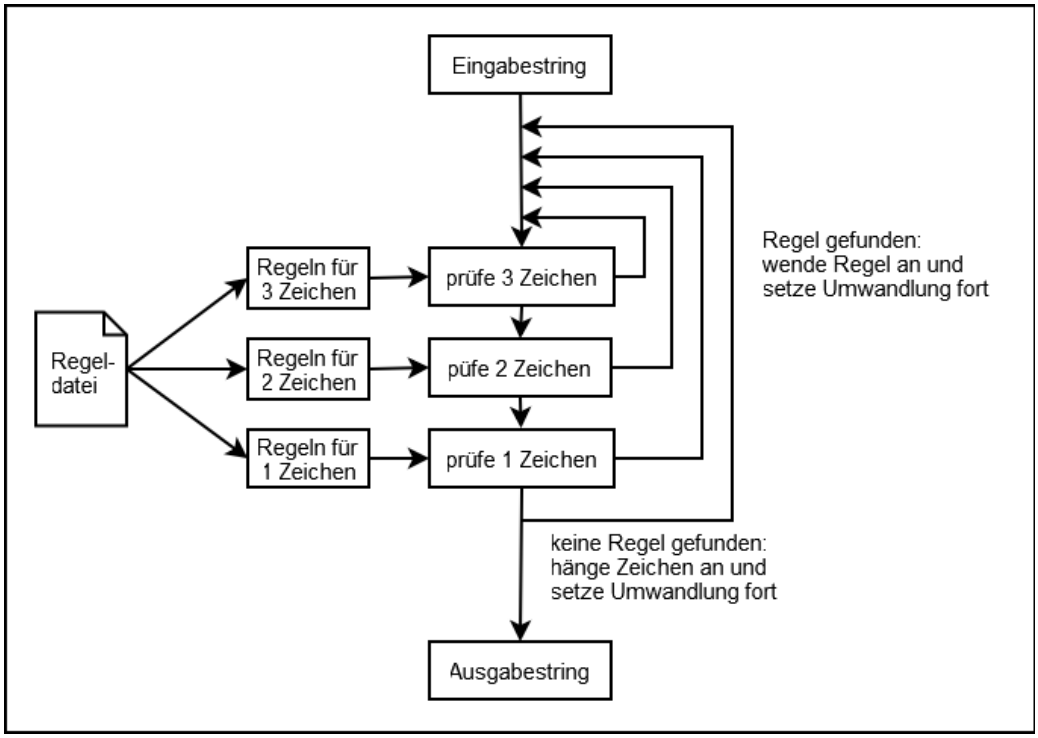

Abbildung 13: Schema der phonetischen Transformation

Um die Regeln übersichtlicher zu gestalten wurden verschiedene Funktionen definiert, etwa boolesche Operatoren und Makros (z.B. eine Regel für sämtliche Vokale).

Die Regelmenge bestimmt zum großen Teil auch die Genauigkeit der phonetischen Suche. Werden viele Grapheme auf dieselben Phonemklassen abgebildet, so reduziert sich auch die Anzahl der phonetischen Wortklassen, sprich die Anzahl unterscheidbarer Worte. Die Suche liefert dann mehr falsch-positive Ergebnisse. Werden jedoch nur wenige Grapheme auf gleiche Phonemklassen abgebildet, so ist der Unterschied zum Ergebnis der einfachen Suche nur minimal. Genau diesen Ansatz verfolgt das zweistufige Verfahren.

Die erste Stufe

In der ersten Stufe werden nur unhörbare Varianten (z.B. Doppelkonsonanten oder verschiedene Grapheme für ein Phonem) umgewandelt. Dies wird durch fast 40 Regeln erreicht. Das Ergebnis ist ein nach wie vor griechisches Wort, welches genauso klingt wie das Original.

Diese Stufe eignet sich zur Kompensation kleinerer Unsicherheiten bei der Schreibung des Suchbegriffes und ist daher für einen geübten Benutzer oft ausreichend. Bei größeren Unsicherheiten (z.B. durch undeutliche oder zu schnelle Aussprache) muss das Verfahren allerdings passen. Die erste Stufe kommt dem Soundex-Algorithmus am nächsten, ohne jedoch so grob zu klassifizieren wie dieser.

Die zweite Stufe

In der zweiten Stufe wird eine Reduktion der Stichworte auf ausgewählte Phonemklassen des Neugriechischen durchgeführt. Dazu werden die Grapheme der Stichworte bzw. des Suchbegriffes durch die Regeln den Phonemklassen zugeordnet. Am Ende wird somit dem Eingabewort seine phonetische Beschreibung als Folge von Phonemklassen zugewiesen. Dabei wird vorausgesetzt, dass die Umwandlung der Stufe 1 bereits geschehen ist. Die zweite Stufe umfasst etwas mehr als 40 Regeln.

Am Ende des Umwandlungsprozesses kann es passieren, dass doppelte Zeichen in der phonetischen Beschreibung auftreten. Diese werden zum Abschluss eliminiert. Außerdem gibt es Alternativen in einigen Regeln, die somit mehrere mögliche phonetische Beschreibungen zu einem Eingabewort erzeugen. Die phonetische Umwandlung ist also nicht immer eindeutig.

Ein Spezialfall tritt ein, wenn der Benutzer die Suchoption „Teilwort“ aktiviert. Dann kann es passieren, dass der Suchbegriff endet, bevor eine Regel für zwei oder drei Buchstaben vollständig abgearbeitet werden konnte. Damit dadurch keine korrekten Treffer entfallen, wird die Regel als *wahr* angenommen. Darüber hinaus kann es vorkommen, dass mehrere gleichberechtigte Regeln partiell am Ende des Suchbegriffes zutreffen. Diese werden dann alle als *wahr* angenommen und führen damit zu mehreren möglichen phonetischen Beschreibungen.

Der Algorithmus ist auch auf andere Sprachen und Alphabete übertragbar. Zur Demonstration der Flexibilität wurden die Regeln so gestaltet, dass sich griechische Suchbegriffe mit lateinischen Buchstaben, die dem vernommenen Klangbild im Deutschen entsprechen, eingeben lassen. Aus diesem Grund gibt es nicht nur Regeln für griechische sondern ebenso für lateinische Buchstaben, was die Anzahl der Regeln fast verdoppelt.
Die zweite Stufe ist immer dann gefragt, wenn eine große Unsicherheit bei der Schreibung des Suchbegriffes herrscht. Sie kann als letzte Variante gewählt werden, wenn alle anderen Suchverfahren scheitern.

4.4. Kombination: Stammsuche und phonetische Suche

Eine Kombination aus Stammsuche und phonetischer Suche ist durchaus sinnvoll, da die Aussprache eines Wortes von seiner Flexionsendung abhängen kann, die Endung beeinflusst also die Aussprache des Stammes.
Sofort fällt eine triviale Methode der Kombination auf: die Hintereinanderausführung beider Suchverfahren. Entweder es wird eine phonetische Suche auf die „Roots“ bzw. Stämme des Suchbegriffes angewendet oder es wird eine Stammsuche auf den phonetischen Beschreibungen bzw. Suchbegriffen durchgeführt. Spezielle Regeln, die damit noch nicht abgedeckt werden, sind bislang jedoch noch nicht implementiert worden.
Der „wbuch“-Data-Mart enthält bereits die phonetischen Entsprechungen aller „Roots“ für beide Stufen der phonetischen Suche. Die normale Hintereinanderausführung von Stammsuche und phonetischer Suche ist also möglich.

4.5. Trefferrelevanz

Für die Bestimmung der Relevanz eines Treffers gibt es verschiedene Kriterien im Wörterbuch.

- Wurde nach mehreren Worten gesucht, so werden die Einträge zuerst nach der Anzahl der enthaltenen Suchbegriffe geordnet.

- Das nächste Kriterium ist der Prioritätswert. Dieser wird bei der Generierung des „wbuch“-Data-Mart für jedes Stichwort ermittelt, auch für die „Roots“. Er richtet sich hauptsächlich nach der Art des Stichwortes, z.B. Grundform, gebeugte Form oder auch „Root“.
- Im „wbuch“-Data-Mart enthalten, aber bislang noch ungenutzt, ist die Angabe der Position eines Stichwortes innerhalb eines Eingabefeldes. Dies ist vor allem bei Stichworten interessant, welche vom Autor explizit angegeben wurden. Hier lässt sich damit über die Reihenfolge dieser Stichworte die Relevanz beeinflussen.
- Für die phonetische Suche wurde der Edit-Distance-Wert eingeführt. Dieser beschreibt grob die Anzahl der Umwandlungsschritte vom Eingabewort zur phonetischen Beschreibung. Je geringer die Differenz der Edit-Distance-Werte zweier Worte ist, desto ähnlicher sind sie, und desto höher ist die Relevanz eines Treffers. Die Wertigkeit der einzelnen Regeln kann in der Regeldatei genau angegeben werden, da manche Regeln mehrere Buchstaben verändern und andere nur einen.
- Wenn alle Kenngrößen ausgewertet wurden, so werden die dann noch gleichermaßen relevanten Einträge ganz normal alphabetisch sortiert, entweder nach dem griechischen oder nach dem deutschen Alphabet.

Die bei einer Suche herangezogenen Relevanzkriterien richten sich nach dem verwendeten Suchverfahren. Spätere Erweiterungen sind dabei durch die flexible Architektur jederzeit möglich.

5. Zusammenfassung

Das vorgestellte Wörterbuch überwindet die Barriere zwischen verschiedenen Sprachen und Zeichensätzen. Der aktuelle Stand der verfügbaren Datenbanken und Programmiersprachen erlaubt die Realisierung komplexer mehrsprachiger Anwendungen ohne grundsätzliche Probleme. Sie können jedoch immer noch in Detailfragen, z.B. bei nicht weit verbreiteten Sprachen, auftreten.

Das Modell des Wörterbuches erlaubt detaillierte Angaben zu allen Einträgen und ist sehr flexibel. Grammatikalische Einstellungen lassen sich zentral anpassen. Für eine performante Umsetzung der vorgestellten Suchstrategien wurde mit Data-Warehouse-Techniken ein spezielles Konstrukt, der „wbuch“-Data-Mart, erschaffen. Außerdem ist durch eine integrierte Kommentarfunktion die Mitarbeit aller Benutzer möglich, um den Umfang des Wörterbuches stetig zu vergrößern.

Die Übernahme der bestehenden Daten aus dem Vorgänger verursachte zunächst Probleme. Eine lückenhafte Dokumentation und fehlerhafte Einträge erschwerten die Konvertierung in das neue Modell, nach einer teilweise aufwendigen manuellen Analyse gelang dies jedoch. Gefundene Fehler wurden protokolliert und später von den Autoren abgearbeitet.

Die Stammsuche erkennt verwandte Worte zu Suchbegriffen, die selbst nicht im Wörterbuch stehen. Die phonetische Suche kann griechische Worte an ihrer Aussprache erkennen, selbst wenn diese mit deutschen Buchstaben eingegeben werden. Die Regeln und Grundlagen dieser mächtigen Suchverfahren sind zentral in Textdateien gespeichert und doch flexibel zugleich. Sogar die Kombination beider Strategien ist möglich, jedoch muss hier noch intensiv an einer Verbesserung des Algorithmus gearbeitet werden, da momentan lediglich eine Hintereinanderausführung der beiden Suchstrategien implementiert ist. Für die

Relevanzbestimmung der Treffer werden verschiedene Kriterien und Kenngrößen herangezogen, z.B. ein Prioritätswert und ein Edit-Distance-Wert.

Zukünftig sind noch verschiedene Erweiterungen des Wörterbuches möglich, z.B. die Angabe der Lautschrift zu einem Eintrag mittels Internationalem phonetischen Alphabet (IPA). Auch ein Im- und Export per XML eröffnet weitere Möglichkeiten.

Durch Hinzufügen weiterer Schnittstellen (alternativ zum „wbuch“-Data-Mart) können auf Basis der vorliegenden Daten noch vielfältige Anwendungen implementiert werden: Vokabeltrainer, Rechtschreibhilfen oder Wortschatzanalysen. Die Datenstrukturen und Verfahren des Wörterbuches lassen sich auch bei ähnlich gelagerten Aufgaben in anderen indogermanischen Sprachen einsetzen.

A. Quellenverzeichnis

[Wi06] Wikipedia: Teiresias (http://de.wikipedia.org/wiki/Teiresias)

[Un06] The Unicode Consortium: What is Unicode? (http://www.unicode.org)

[My06] MySQL AB: MySQL 5.0 (http://www.mysql.com)

[Ja06] Sun Microsystems: J2SE 1.4 (http://java.sun.com)

[PHP06] The PHP Group: PHP 5 (http://www.php.net)

[HBL83] K. Heß, J. Brustkern, W. Lenders: Maschinenlesbare deutsche Wörterbücher, 1983, Niemeyer-Verlag

[Sch87] B. Schaeder: Germanistische Lexikographie, 1987, Niemeyer-Verlag

[Vo99] M. Volk: Lexikonaufbau und Morphologie-Analyseverfahren, 1999, Universität Zürich

[Ru02] H. Ruge: Die Grammatik des Neugriechischen, 2002, Romiosini-Verlag

[PO05] PONS: Kompaktwörterbuch Neugriechisch, 2005, Klett-Verlag

[Gl67] H. Glinz: Der Deutsche Satz, 1967, Schwann-Verlag

[Io92] Anna Iordanidou: Τα ρήματα της νέας Ελληνικής (Die Verben des Neugriechischen), 1992, Pataki-Verlag

[Tr99] M. Triantafyllidis: Λεξικό της κοίνης νεοελληνικής (Lexikon des Neugriechischen), 1999, Universität Thessaloniki - Manolis Triantafyllidis Stiftung

[Ra06] E. Rahm: Datenbanksysteme I, 2006, Universität Leipzig

[Le05] H. Müller, M. Weis, J. Bleiholder, U. Leser: Erkennen und Bereinigen von Datenfehlern in naturwissenschaftlichen Daten, 2005, Datenbank-Spektrum 15/2005

[ADMR05] Aumueller, David; Do, Hong-Hai; Massmann, Sabine; Rahm, Erhard: Schema and Ontology Matching with COMA++, 2005, SIGMOD 2005

[Si03] Brijesh Shanker Singh: Search Algorithms, 2003, Indian Statistical Institute

[So07] D. Sosna: Phonetische Suche in neugriechischen Texten, 2007, Universität Leipzig

B. Abbildungsverzeichnis

C. Tabellenverzeichnis

D. Inhalt der CD

Die beigelegte CD enthält die Implementierung der Web-Oberfläche und des Eingabeprogramms inklusive aller Dokumentationen sowie Hinweise zur Installation. Der Vollständigkeit halber ist auch das Programm zur Konvertierung der Vorgängerdaten dabei, auch wenn es nicht mehr auf dem aktuellen Stand des Datenbankschemas ist.
Weiter befinden sich auf der CD die SQL-Statements zum Erstellen der Datenbank, eine ausführliche Beschreibung inklusive einiger Testdatensätze, sowie alle zum Betrieb notwendigen grammatikalischen Daten.
Die Datei „Readme.txt“ im Wurzelverzeichnis der CD enthält außerdem weitere Informationen über die Software (Datenbank, Webserver usw.), die für die Benutzung des Wörterbuches notwendig ist.

E. Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

_____	_____	_____
Ort	Datum	Unterschrift