Universität Zürich

Computerlinguistik

e-GramD
E-Learning mit Hilfe von computerlinguistisch annotierten Übungstexten für den Deutschunterricht auf Primarstufe

Lizenziatsarbeit an der Philosophischen Fakultät der Universität Zürich

Referent: Prof. Dr. Michael Hess

Verfasserin:

Sandra Lüthi

Albisstrasse 28

8038 Zürich

Zusammenfassung

Es gibt zwei Arten von webbasierten CALL-Anwendungen (Computer Assisted Language Learning): traditionelle und intelligente CALL-Applikationen (ICALL). Traditionelle Anwendungen können nur für vordefinierte Lösungen Feedback geben. ICALL-Applikationen dagegen arbeiten mit NLP-Technologien (Natural Language Processing) und können dadurch je nach Fehler unterschiedliche und intelligente Rückmeldungen bieten. Der Nachteile von ICALL-Applikationen ist die Tatsache, dass die Datenanalysen von NLP-Tools nicht fehlerfrei sind.

Aus diesem Grund stellt die vorliegende Arbeit eine E-Learning-Applikation (*e-GramD*) vor, die Übungen auf der Basis eines Korpus von annotierten Texten aufbaut. Mit Hilfe eines Parsers wird ein Korpus von Übungstexten analysiert, deren Annotationen manuell überprüft werden. *E-GramD* ist für den Deutschunterricht an der Grundschule gedacht und wurde von sechzehn Schülern und vier Lehrpersonen dieser Unterrichtsstufe getestet.

Die vorliegende Arbeit zeigt, dass die Einbindung von annotierten Daten einen entscheidenden Vorteil gegenüber üblichen ICALL-Anwendungen bietet, ohne dass auf sinnvolle Rückmeldungen verzichtet werden muss. Die manuelle Überprüfung der geparsten Sätze garantiert korrekte Rückmeldungen, weil fehlerhafte Annotationen auf diese Weise aufgefangen werden können.

Abstract

There are two types of web-based CALL (Computer Assisted Language Learning) applications: traditional and intelligent CALL applications (ICALL). Most of the traditional CALL applications can only provide feedback for predefined solutions. ICALL systems, however, work with NLP (Natural Language Processing) technologies and thus can give situational and intelligent feedback. The main disadvan-

tage of ICALL applications is the fact that NLP-tools do not provide entirely correct analyses.

For this reason, this thesis presents an E-Learning application (*e-GramD*) that builds excercises based on a corpus of annotated texts. The corpus of excercise material is first analysed by a NLP parser and then checked manually to eliminate parse errors. *E-GramD* is built for German classes at primary schools and has been tested by sixteen pupils and four teachers of this particular level.

The present work shows that there is one distinct advantage by integrating annotated data, without abandoning intelligent feedback, over other ICALL systems. Since all parsed sentences are manually checked, incorrectly annotated data and thus erroneous feedback can be avoided.

Inhaltsverzeichnis

Zusammenfassung i

Abstract i

Inhaltsverzeichnis iii

Abbildungsverzeichnis v

Tabellenverzeichnis vi

Danksagung vii

1 Einleitung 1

1.1 Ausgangslage 1

1.2 Ziel und Aufbau der Arbeit 2

2 Neue Medien im Unterricht 5

2.1 Begrifflichkeit der neuen Medien 5

2.1.1 Von Plato zum e-Tutor 7

2.1.2 PLATO 7

2.1.3 The Bridge German Tutor 8

2.1.4 e-Tutor 10

3 Usability und CALL-Anwendungen 12

3.1 Usability-Standards 13

3.2 Anwendungsqualität (quality in use) 15

3.3 Wartbarkeit (maintainability) und Autorenprogramme 20

3.4 Verlässlichkeit (reliability) bei fehlerhaftem Input 23

4 Grammatikunterricht Deutsch 29

4.1 Unterrichtsformen und Lerntheorien 29

4.1.1 Behaviorismus 30

4.1.2 Kognitive Ansätze 32

4.1.3 Konstruktivistische Ansätze 33

4.2 Grammatikunterricht auf Primarstufe 34

4.2.1 Morphologische Kompetenzen 36
4.2.2 Syntaktische Kompetenzen 38
4.2.3 Semantische Kompetenzen 39
5 ICALL- und CALL-Anwendungen im Vergleich 41
5.1 Kriterienkatalog 41
5.2 Datensammlung und -erhebung 45
5.3 Diskussion der erhobenen Daten 46
5.3.1 Untersuchte Anwendungen im Kontext 47
5.3.2 Anwendungsqualität der untersuchten Applikationen 51
6 e-GramD 58
6.1 NLP 58
6.2 Systemarchitektur und Eigenschaften von e-GramD 61
6.3 Modul zum Erstellen und Bearbeiten von Lerninhalten 64
6.4 Modul zum Lösen von Übungsaufgaben 69
6.4.1 Anklickübungen 71
6.4.2 Lückentexte 73
7 Evaluation e-GramD 77
7.1 Evaluation von E-Learning-Anwendungen: Grundsätzliche Überlegungen 78
7.2 Usability von e-GramD 81
7.3 Resultate Evaluationsphase Schüler 84
7.4 Resulate Evaluationsphase Lehrpersonen 88
7.5 Synthese der drei Evaluationsteile 90
8 Zusammenfassung und Ausblick 93
Literaturverzeichnis 97
Appendix A 101
Appendix B 104
Appendix C 105
Curriculum Vitae 106

Abbildungsverzeichnis

Abbildung 1: Komponenten eines ITS 9

Abbildung 2: User-orientiertes Software Design 14

Abbildung 3: Software-Qualitätskriterien 15

Abbildung 4: Rückmeldung Schultraining 26

Abbildung 5: Rückmeldung Lernareal 27

Abbildung 6: Rückmeldung e-Tutor 28

Abbildung 7: Systemarchitektur e-GramD 62

Abbildung 8: Übersicht Übungstexte 65

Abbildung 9: Einlesen eines neuen Textes 66

Abbildung 10: Korrektur eines geparsten Textes 68

Abbildung 11: Einstiegsseite Übungen 70

Abbildung 12: allgemeine Auswertung nach Korrektur 71

Abbildung 13: detaillierte Rückmeldung Anklickübung 72

Abbildung 14: Feedback Rechtschreibefehler Lückentexte 75

Abbildung 15: Feedback Deklinationsfehler Lückentexte 75

Abbildung 16: Konzept der Lernsoftware-Evaluation EvaSoft 79

Tabellenverzeichnis

Tabelle 1: Flexionsklassen des Deutschen 36

Tabelle 2: Kriterienkatalog CALL Anwendungen 42

Tabelle 3: Überblick evaluierte Anwendungen 46

Tabelle 4: traditionelle CALL-Anwendungen im Kontext 47

Tabelle 5: ICALL-Anwendungen im Kontext 49

Tabelle 6: Anwendungsqualität von traditionellen CALL-Anwendungen 51

Tabelle 7: Anwendungsqualität von traditionellen CALL-Anwendungen 53

Tabelle 8: Übersicht über morphologische Parser 59

Tabelle 9: Übersicht Module von e-GramD 82

Tabelle 10: Übersicht Evaluation Schüler 86

Danksagung

Ich bedanke mich bei Dr. Manfred Klenner für die Unterstützung dieser Lizenziatsarbeit, obwohl das Thema von den traditionellen Unterrichtsthemen der Computerlinguistik der Universität Zürich abweicht. Den vier involvierten Lehrpersonen danke ich für ihre praxisbezogenen und pädagogisch wertvollen Hinweise, was den Einsatz von e-Learning Anwendungen im Unterricht anbelangt. Ein besonderer Dank gilt der vierten Klasse aus Nussbaumen (AG), die *e-GramD* ausführlich getestet und mir konstruktives Feedback gegeben hat.

Connexor Oy danke ich für die Evaluationslizenz von *Machinese Syntax*. Bei Dr. Trude Heift bedanke ich mich für das Einrichten einer *e-Tutor*-Probelizenz und für ihre Unterstützung bei meinen Fragen zu dieser Software. Bei Csilla Puskás bedanke ich mich für die zusätzlichen Informationen zu ihrem Prototyp *Robi*.

Torr Brown und David Phillips danke ich für die Tier- und Landschaftsfotos, die als Hintergrundbilder von *e-GramD* verwendet werden. David Phillips danke ich ausserdem herzlich für die programmiertechnischen Ratschläge und Anregungen während der Entwicklungsphase. Ein besonderes Dankeschön gilt der Korrektorin Myrta Lüthi, die meine Arbeit sprachlich, sowie meinen Kommilitonen Angela Fahrni und Christian Hardmeier, die sie inhaltlich hinterfragt haben.

1 Einleitung

1.1 Ausgangslage

Der aargauische Lehrplan für die Primarstufe, also 1.-5. Klasse, stellt für den Grammatikunterricht das Erforschen der Sprachstruktur in den Vordergrund (Aarg. Lehrplan, 2005). Obwohl der aargauische Lehrplan softwareunterstütztes Lernen in den Bereichen Lese-, Schreib- und Orthografieunterricht empfiehlt, gibt es keine Hinweise auf solche Anwendungen für den Grammatikunterricht. Gerade aber für diesen sind oft zusätzliche Übungen nützlich, um das Erlernte zu vertiefen. CALL-Applikationen (Computer Assisted Language Learning) bieten den Vorteil, dass die Lernenden in ihrem eigenen Tempo auch ausserhalb des Präsenzunterrichts lernen können und eine Lernkontrolle für die Schüler unabhängig von der Lehrperson möglich ist. Die Lernenden erhalten unmittelbar nach Abschluss der Aufgabe ein sachliches Feedback. Ausserdem bieten Übungen am Computer eine Abwechslung zum traditionellen Unterricht und werden daher von den Schülern gerne gelöst.

Im Internet sind diverse frei verfügbare E-Learning[1]-Anwendungen für den Deutschunterricht zu finden, diese weisen aber verschiedene Mängel auf. Ein Hauptproblem sind die Rückmeldungen, die oft nicht sehr aufschlussreich sind, weil sie nicht auf didaktischen, sondern auf programmiertechnischen Überlegungen beruhen. Wenn die Lernenden Verben in einem Text suchen sollen und sie nebst drei Verben noch zwei Adjektive anklicken, hilft ihnen das Feedback «Das

1 Die Bezeichnung E-Learning bezieht sich ausschliesslich auf internetbasiertes Lernen.

ist teilweise richtig»[2] nicht weiter. Weder wurden die beiden falsch angeklickten Adjektive lokalisiert noch wurde eine Fehleranalyse ausgegeben.

ICALL-Anwendungen (Intelligent Computer Assisted Language Learning) bedienen sich der Techniken der Computerlinguistik und der künstlichen Intelligenz. Eine Eingabe wird z.B. mit Hilfe einer Grammatik und eines Parsers verarbeitet, so dass Fehler nicht nur lokalisiert werden, sondern auch Parseinformationen die Analyse des Fehlers unterstützen. So erkennt eine ICALL-Anwendung wie im obigen Beispiel, dass nebst den drei Verben noch zwei Adjektive angeklickt wurden. ICALL-Applikationen können deshalb inhaltlich sinnvolles Feedback ausgeben. Ein grosser Nachteil von ICALL-Applikationen besteht darin, dass auch moderne Parser nicht jeden Satz korrekt analysieren. Das bedeutet, dass zum Teil Satzteile falsch annotiert werden, diese fehlerhaften Informationen anschliessend in die automatisch generierten Feedbacks einfliessen. Schüler erhalten so fehlerhafte Rückmeldungen. Aus diesem Grund werden ICALL-Anwendungen selten im Unterricht eingesetzt und von den Lehrern weniger akzeptiert als traditionelle CALL-Software.

Eine ideale E-Learning-Anwendung nutzt also die Vorteile von computerlinguistischen Methoden, stellt aber ein Werkzeug für die Überprüfung der Parsedaten zur Verfügung.

1.2 Ziel und Aufbau der Arbeit

Ziel der vorliegenden Arbeit ist einerseits ein theoretischer Überblick über CALL und andererseits eine praktische Umsetzung einer E-Learning-Anwendung für den Deutschunterricht auf Primarstufe (*e-GramD*). Für *e-GramD* wird mit Hilfe eines syntaktischen Parsers[3] ein Korpus von Übungstexten analysiert, deren An-

2 http://www.schultraining.ch

3 Machinese Syntax von Connexor

notationen manuell über ein spezielles GUI (Graphical User Interface) überprüft werden. Lehrpersonen können mühelos Übungsmaterial zu den jeweils behandelten Themen im Unterricht einlesen. Die Übungen werden anschliessend auf der Basis der annotierten Übungstexte generiert. Auf diese Weise kann sinnvolles und intelligentes Feedback ausgegeben werden. Gleichzeitig wird vermieden, dass falscher Output des Parsers in die Rückmeldungen einfliesst.

Das vorliegende E-Learning-Tool wurde in Zusammenarbeit mit einer vierten Klasse entwickelt. Während der Entwicklungsphase hat mich die Lehrerin mit pädagogisch wertvollen Input unterstützt. Ausserdem wurde *e-GramD* von sechzehn Schülern und vier Lehrpersonen dieser Stufe getestet und evaluiert.

Die Arbeit ist in sechs Themenbereiche aufgegliedert. **Kapitel 2** klärt den technischen und historischen Hintergrund von CALL und E-Learning. **Kapitel 3** beschreibt die Qualitätsanforderungen an solche Lehrmittel, die erfüllt sein müssen, damit diese im Unterricht tatsächlich integriert werden können. **Kapitel 4** führt drei didaktische Theorien zur Gestaltung des Unterrichts ein und betrachtet den Deutschunterricht auf Primarstufe genauer. Die Analyse bestehender CALL-Anwendungen für das Erlernen von Deutsch ist der Fokus von **Kapitel 5**. Dabei soll ein Kriterienkatalog erarbeitet werden, mit dem zwölf bestehende Anwendungen auf technische und medienpädagogische Merkmale untersucht werden. **Kapitel 6** stellt die Systemarchitektur von *e-GramD*, den verwendeten Parser und die verschiedenen Funktionalitäten zum Erstellen und Lösen der Übungen im Detail vor. **Kapitel 7** knüpft an den Kriterienkatalog von Kapitel 5 an, analysiert *e-GramD* aufgrund dessen und präsentiert die Resultate der beiden Evaluationen (Schüler und Lehrpersonen). Schliesslich greift **Kapitel 8** die Resultate der Arbeit nochmals auf und zeigt die Konsequenzen, welche für die Ent-

wicklung von computerlinguistisch motivierten E-Learning-Anwendungen für den Deutschunterricht gezogen werden können.

2 Neue Medien im Unterricht

Es ist nichts Neues, dass technische Hilfsmittel, meist als Medien bezeichnet, im Unterricht eingesetzt und mit Lernen in Verbindung gebracht werden (vgl. Baumgartner 2002, Kerres 2001 und Moser 1999). Seit den sechziger Jahren kommen auch sogenannte *neue Medien*, also Computer, in der Schule zum Einsatz. Dieses Kapitel beschäftigt sich mit dem technischen Hintergrund von neuen Medien und klärt zentrale Begriffe (Abschnitt 2.1). Ausserdem wird in Abschnitt 2.2 anhand von drei ausgewählten Anwendungen ein historischer Abriss der neuen Medien vorgenommen.

2.1 Begrifflichkeit der neuen Medien

Ein Medium kann allgemein als ein Hilfsmittel definiert werden, das einen Inhalt zwischen einer (oder mehreren) kommunizierenden und einer (oder mehreren) adressierten Person transportiert (Burger, 2005:32). Medien unterstützen Lehrpersonen in ihrer Tätigkeit, indem sie organisieren und die Lerninhalte für die Lernenden veranschaulichen. Die Wahl der einzusetzenden Medien hängt stark vom zu vermittelnden Lernziel ab, z.B. wird für die Aneignung von Hörverstehen ein Kassettenrecorder eingesetzt. Mitschian (1999:87) unterscheidet zwischen zwei Medientypen: schon lange im Unterricht werden *traditionelle Medien* wie Wandtafeln verwendet. Unter *modernen Medien* werden alle Hilfsmittel verstanden, die elektrisch betrieben sind, also Hellraumprojektor oder Kassettenrecorder. Innerhalb der modernen Medien werden alle computergestützten Geräte zur Sondergruppe *neue Medien* gezählt (Mitschian, 1999:87). Eine Anwendung

gilt dann als *multimedial,* wenn verschiedene Medien auf einem einzigen Medienträger verknüpft sind (Mitschian, 1999:74), z.B. ein Tonbandgerät in Verbund mit einem Diaprojektor. Anwendungen, die für neue Medien konzipiert sind, können verschiedene Medienformate (visuell, auditiv etc.) relativ einfach integrieren. Deshalb finden sich heute sehr viele multimediale Lehrmittel für die neuen Medien.

Heute ist der Einsatz neuer Medien nicht mehr aus dem Unterricht wegzudenken, wenngleich anfänglich unklar war, wie sie pädagogisch sinnvoll in den Unterricht integriert werden sollten. Es waren vor allem technisch begeisterte Lehrpersonen, die verschiedene Anwendungsmöglichkeiten für diese Art Medien ausprobierten. Es ist daher nicht verwunderlich, dass immer neue Fachbegriffe je nach Einsatzmöglichkeit und Hintergrund der Anwendungen geschaffen wurden. Computergestütztes Lernen im Unterricht wird je nach Autor und Zeit als *CBT* (Computer Based Training), *CAI* (Computer Assisted Instruction)[4] und *CAL* (Computer Assisted Learning) bezeichnet (vgl. Chambers 1983, Kerres 2001 und Schulmeister 2002). Der Begriff *CALL* (Computer Assisted Language Learning) hat sich durchgesetzt, wenn sich der Lerninhalt, der für den Unterricht aufbereitet wird, auf den Sprachunterricht beschränkt. *Intelligent CALL* bezeichnet CALL-Anwendungen, die Methoden und Techniken der künstlichen Intelligenz einsetzen. Dabei kann eine Anwendung z.B. *NLP* (Natural Language Processing) einsetzen und mit Hilfe eines Lexikons und einer Grammatikprüfung eine Eingabe evaluieren.

Seit den neunziger Jahren und mit der rasanten Verbreitung des Internets sind zunehmend auch *WBT*-Anwendungen (Web Based Training) im Unterricht zu fin-

4 In den Vereinigten Staaten wird CAI oft als CBI (computer-based instruction) oder CBE (computer-based education) bezeichnet.

den. Heute wird *E-Learning* dem Begriff WBT vorgezogen, obwohl E-Learning unterschiedlich definiert wierd und keine eindeutige Begriffsbestimmung vorherrscht. Je nach Informationsquelle wird E-Learning als Sammelbegriff für alle Formen elektronisch unterstützten Lernens (Baumgartner 2002:302) oder als Oberbegriff für alle Formen internetbasierter Lehr- und Lernangebote (Kerres, 2001:14) verstanden. In der vorliegenden Arbeit bezieht sich die Bezeichnung E-Learning ausschliesslich auf internetbasiertes Lernen. E-Learning hat den Vorteil, dass es sowohl zeitlich wie auch örtlich flexibel eingesetzt werden kann. Wenn eine E-Learning-Anwendung in den traditionellen Präsenzunterricht integriert wird, spricht man vom *Blended Learning* (Rösler, 2004:10).

2.1.1 Von Plato zum e-Tutor

Der Einsatz neuer Medien im Sprachunterricht ist nicht erst ein Phänomen der letzten Jahre. Seit den sechziger Jahren wird im Bereich von CALL geforscht und es werden Anwendungen programmiert. *PLATO* ist ein prototypisches System aus den sechziger Jahren, der *Bridge German Tutor* ein exemplarisches Beispiel der *ITS*-Forschung (Intelligent Tutoring System) der achtziger Jahre und der *e-Tutor* gehört der neusten Generation von webbasierten ICALL-Applikationen an.

2.1.2 PLATO

Bereits in den sechziger Jahren wurde ein Mainframe-System an der Universität Illinois unter dem Namen *PLATO* (Programmed Logic for Automated Teaching Operations) entwickelt, das den Studierenden ermöglichen sollte, in ihrem eigenen Tempo zu lernen (Chambers, 1983:8). *PLATO* war ein multimediales System und konnte unabhängig von der Fachrichtung eingesetzt werden. Es konnten nebst Texten auch Grafiken, Farbfotografien und Diagramme auf Plasmabildschirmen angezeigt oder Sprache über einen Magnetplattenspieler aufgenommen

oder wiedergegeben werden (Mitschian, 1999:51f). Die Übungen basierten auf dem Prinzip von B. Skinner (siehe Abschnitt 4.1), das Feedback beschränkte sich auf die Antworten «richtig» oder «falsch». In den siebziger Jahren wurde *PLATO* in über 50'000 Stunden von Studierenden an der Universität Illinois benutzt, und das System sollte vermarktet werden. Das Projekt *PLATO* wurde aber bald nach diesem Entscheid eingestellt, weil die Anschaffungskosten für das Mainframe-System in der Höhe von einigen Millionen Dollar lagen. Gleichzeitig wurden andere günstigere Lösungen für Mikrocomputer angeboten, auch wenn diese nicht mehr multimedial waren, da die Rechenleistung der Mikrocomputer dazu viel zu gering war. Erst die technischen Fortschritte der neunziger Jahre eröffneten die Möglichkeit für Mikrocomputer multimediale Anwendungen zu programmieren, inzwischen allerdings «auf einem technisch wesentlich höheren Niveau» (Mitschian, 1999:55) als bei *PLATO*. Heute werden Kurse, die auf dem ursprünglichen *PLATO*-System basieren, von *PLATO* Learning[5] auch für PCs verkauft.

2.1.3 The Bridge German Tutor

In den achtziger Jahren fokussierte sich die Forschung auf ITS (Intelligent Tutoring System), ICALL-Systeme, die Anweisungen und Rückmeldungen flexibel dem Niveau der Schüler anpassen können. Ein ITS besteht aus drei Hauptkomponenten: einem Experten-, einem Studenten- und einem Instruktorenmodell. Das Expertenmodell enthält das Wissen, das vermittelt werden soll. Das Studentenmodell repräsentiert den Wissenstand der Schüler und wie sie eine Übung lösen. Das Instruktorenmodell basiert auf einer bestimmten Unterrichtsstrategie, die vorgibt, wie der Inhalt vermittelt werden soll. Auf diese Weise kann ein ITS das Übungsprogramm auf die Schüler entsprechend deren Wissenstand abstim-

5 http://www.plato.com/

men, Hinweise geben, wie eine Übung zu lösen ist, und ein lehrreiches Feedback geben.

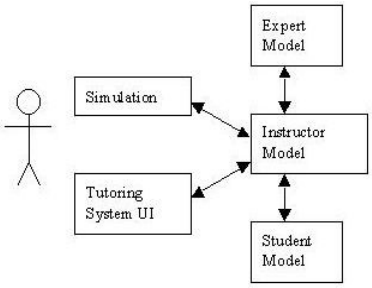


Abbildung 1: Komponenten eines ITS (Ong, 2000:ohne Seitenzahl)

Ein erfolgversprechendes ITS war der *Bridge German Tutor* des Army Research Institute (ARI) (vgl. Sams, 1995). Die Entwicklung des *Bridge German Tutors* wurde von der amerikanischen Armee finanziert und ist daher auf die Bedürfnisse von Soldaten zugeschnitten. Das zeigt sich z.B. im Expertenmodell, welches aus 5'000 Wörtern besteht, die in militärischen Alltagssituationen benützt werden. Für den *Bridge German Tutor* wurde ein robuster Parser implementiert, der auf einer *GB*-Grammatik (Government and Binding) basiert (Sams, 1995:10). Das System kann mit Satzfragmenten umgehen, wenn auch nicht in allen Fällen. Über ein Studentenmodell kann der *Bridge German Tutor* die Übungsabfolge dem Niveau des Lernenden anpassen. Der *Bridge German Tutor* bietet verschiedene Übungstypen an: von Multiple-Choice-Übungen bis hin zu freier Antwort durch den User auf eine bestimmte Frage des Systems. Die eingeschränkten Feedbackmöglichkeiten von *PLATO* wurden von den Autoren des Bridge Tutors verfeinert und ausgebaut. So gibt der *Bridge German Tutor* weder einfache Ja/Nein-Antworten noch kryptische Meldungen in der Art «preposition + case error: bei, das_tal» aus, wie das zu Beginn der Entwicklung der Fall war (Sams, 1995:19), sondern natürlichsprachliche Rückmeldungen wie zum Beispiel «The

preposition bei [dat] does not agree with das Tal» (Sams, 1995:15). Seit 1995 ist keine Literatur zu diesem System mehr erschienen.

2.1.4 *e-Tutor*

Der Trend der neunziger Jahre ging dahin, bestehende CALL-Systeme internetfähig und multimedial zu machen. Der Vorteil des Internets ist, dass Schüler von irgendeinem Computer auf das CALL-System zugreifen können, vorausgesetzt, sie verfügen über einen Internetanschluss. Dadurch kann sich allerdings die potenzielle Benutzergruppe vergrössern und somit heterogener werden. Ausserdem werden Leistungfähigkeit und Effizienz ein kritisches Thema (vgl. Abschnitt 3.2). Wahrscheinlich wurden deswegen mehrheitlich traditionelle CALL-Systeme, die auf Drill- und Practice-Übungen[6] beruhen und eingeschränkte Rückmeldungen geben können, im Web implementiert. Es gibt aber auch einige wenige webbasierte ICALL-Systeme, die auf der Idee der ITS beruhen. Der *e-Tutor*[7], eine Nachfolgeversion des German Tutor, ist ein solches ICALL System, welches die speziellen Eigenheiten des Internets berücksichtigt (Heift 2001a, Heift 2001b, Heift 2006). Wie alle ITS besteht auch der *e-Tutor* aus Domänenwissen (HPSG-Grammatik und Parser), Studenten- und Instruktionsmodellen. Um die Verzögerungszeit zwischen Input und Rückmeldung so klein als möglich zu halten, werden zwei Massnahmen getroffen. Erstens beschränken sich die Übungen auf einzelne Sätze. Zweitens wird eine zusätzliche Analysehierarchie dazwischengeschoben, welche die Eingabe z.B. auf Punktuation und fehlende oder zusätzliche Worte überprüft. Wenn der Schüler also einen Interpunktionsfehler macht, wird die Analyse sofort abgebrochen und eine entsprechende Rückmeldung ausgegeben.

6 Drill- und Practice-Übungen bestehen aus kleinen Übungsabfolgen, mit denen bestimmte Regularitäten eingedrillt werden.

7 http://www.e-tutor.org

Der NLP-Parser wird erst aufgerufen, wenn keiner dieser Fehler analysiert wurde. Das Studentenmodell des *e-Tutors* ist sehr umfangreich und hält mit Hilfe individueller Logins die Eingaben aller Anwender fest, wertet diese aus und gibt entsprechend dem Level des Lernenden ein Feedback aus (Heift 2001b:8). Das System *e-Tutor* wird an der Simon-Fraser-Universität in Vancouver im Deutschunterricht angewendet und gleichzeitig weiterentwickelt.

3 Usability und CALL-Anwendungen

Anfänglich waren es hauptsächlich Linguisten, Mathematiker, Kybernetiker und Informatiker, die sich mit der Entwicklung von CALL-Anwendungen beschäftigten und verschiedene utopische Hoffnungen in die neue Lernform setzten. So sollte der Unterricht spannender werden oder die Maschine die Aufgaben der Lehrenden übernehmen, obwohl von vornherein feststand, «dass die Maschine nie die Flexibilität und damit die Leistungsfähigkeit eines menschlichen Helfers – auch ohne fachspezifische oder -didaktische Ausbildung – erreichen wird» (Mitschian, 1999:128). Weder wurden Didaktiker in den Entwicklungsprozess einbezogen noch wurden «usability standards» berücksichtigt (Rösler, 2004:159).

Für den Softwarebereich existieren Normen zur Nutzbarkeit und Bedienerfreundlichkeit, die von der ISO (International Organization for Standardization) ausgegeben werden. Es werden im folgenden zwei ISO-Dokumente[8] genauer betrachtet: ISO/IEC 9126-1, welches Standards beschreibt, die allgemein für die Entwicklung von Software gelten und ISO FDIS 13407, welches sich mit einem user-fokussierten Entwicklungsprozess für interaktive Systeme beschäftigt. Abschnitt 3.1 gibt einen Überblick über die Usability-Richtlinien und führt die für diese Arbeit relevanten Begriffe ein. Wie die Qualität einer Anwendung definiert wird, ist Thema von Abschnitt 3.2. Die Abschnitte 3.3 und 3.4 konzentrieren sich auf zwei Usability-Richtlinien, die eine spezielle Herausforderung für CALL-Systeme sind. In Abschnitt 3.3 wird speziell auf die Wartung von CALL-Anwendun-

8 Die ISO/IEC-Dokumente sind kostenpflichtig, weshalb ich mich nicht auf die Originaldokumente beziehe, sondern auf Nigel Bevans Analysen von 1999 und 2001.

gen mit Hilfe von Autorenprogrammen eingegangen. Die Verlässlichkeit von CALL-Applikationen und wie sie mit fehlerhaftem Input umgehen, ist Thema von Abschnitt 3.4.

3.1 Usability-Standards

Für einen erfolgreichen Einsatz von CALL-Anwendungen im Unterricht ist die Berücksichtigung von technischen und medienpädagogischen Anliegen wichtig. Damit beide Aspekte von Anfang beachtet werden, sollte der Entwicklungsprozess Usability-Standards berücksichtigen. Usability-Richtlinien werden von der Internationalen Organisation für Normung (ISO) festgelegt. Die Usability-Standards beschreiben vier verschiedene Aspekte einer Anwendung (Bevan, 2001:534):

> (1) The use of the product (effectiveness, efficiency and satisfaction in a particular context of use).
> (2) The user interface and interaction.
> (3) The process used to develop the product.
> (4) The capability of an organization to apply user-centred design.

Eine Applikation muss wirkungsvoll, effizient und zufriedenstellend in einem bestimmten Kontext eingesetzt werden können. Eine Voraussetzung dafür ist ein zweckmässiges Interface, mit Hilfe dessen die Anwender mit dem Programm interagieren können. Damit eine solche Anwendung programmiert wird, muss der Anwender ins Zentrum des Entwicklungsprozesses gestellt werden und der Hersteller der Applikation diese Art von Entwicklungsstrategie unterstützen.

Das heisst, es reicht nicht aus, visuell ansprechende und technisch hochstehende CALL-Software zu produzieren, wie dies ursprünglich versucht wurde (und z.T. auch heute noch wird), sondern die zu vermittelnden Lerninhalte müssen dem User, dem angestrebten Ziel und dem Kontext angepasst werden. Bevan

(1999:94) illustriert basierend auf ISO FDIS 13407 (1998) einen Designprozess, der sich ausschliesslich auf die Bedürfnisse der Anwender stützt.

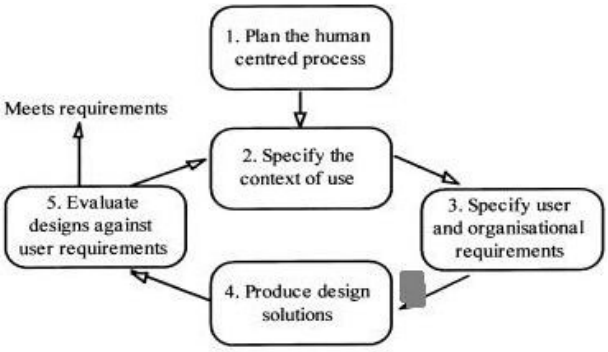


Abbildung 2: User-orientiertes Software Design (Bevan, 1999:94)

Es müssen also vor der Entwicklung eines E-Learning Tools die «inhaltlichen und institutionellen Rahmenbedingungen der möglichen Verwendung» (Kerres, 2001:30) geklärt werden: Welchen Einschränkungen unterliegt der Einsatz des Tools? Was soll damit gelernt werden? Welche Voraussetzungen müssen erfüllt sein, damit das System erfolgreich implementiert werden kann? Dann muss definiert werden, welche Anwender die Zielgruppe der Applikation sind und welche organisationellen Strukturen vorherrschen.

Eine CALL-Anwendung wird von zwei Gruppen benützt, die unterschiedliche Bedürfnisse haben: zum einen von Lehrpersonen, die sich um die Installation der Systeme auf neuen Arbeitsstationen und die Wartung (zumindest des Lerninhaltes) kümmern müssen. Lehrpersonen verlangen von einer Anwendung, dass sie verlässlich und einfach einzuführen ist sowie Funktionalitäten aufweist, die den Unterricht unterstützen. Die zweite Anwendergruppe, die Lernenden, werden mit der Bedienerfreundlichkeit des GUIs konfrontiert. Eine CALL-Applikation muss auf die Bedürfnisse beider Gruppen Rücksicht nehmen. Im Anschluss an die Entwicklung der Anwendung muss eine Evaluation stattfinden, denn nur so

kann festgestellt werden, ob die Bedürfnisse der Anwender vom System erfüllt werden.

3.2 Anwendungsqualität (quality in use)

Eine Anwendung zeichnet sich in erster Linie durch ihre Qualität aus. Betrachten wir diese näher, zeigt sich, dass Anwendungsqualität gemäss ISO/IEC 9126-1 über verschiedene Parameter gemessen (zitiert in Bevan, 2001:537) wird:

> Quality in use: the capability of the software product to enable specified users to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified context of use.

Die Qualität einer Anwendung beruht demzufolge auf sechs Kriterien, zwei davon technischer und vier anwenderspezifischer Natur. Die nachfolgende Abbildung 3 illustriert diese sechs Kriterien.

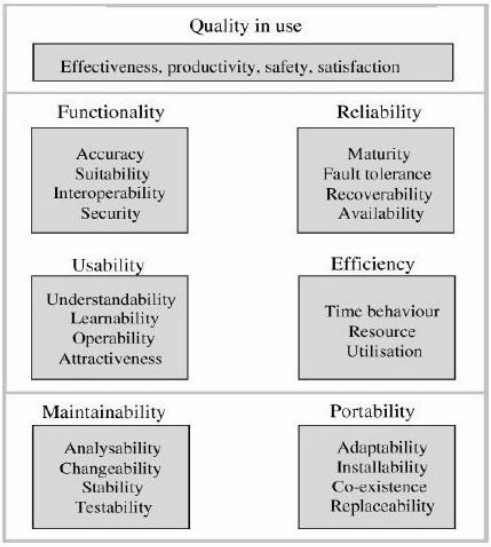


Abbildung 3: Software-Qualitätskriterien (Bevan, 2001:540)

Die technischen Kriterien beziehen sich auf die Übertragbarkeit (portability) und die Wartbarkeit (maintainability). Die anwenderspezifischen Aspekte konzentrieren sich auf die Leistungsfähigkeit (efficiency), die Verlässlichkeit (reliability),

die Funktionalität (functionality) und die Bedienerfreundlichkeit (usability)[9]. Erst das Zusammenspiel aller sechs Punkte garantiert die gute Qualität einer Anwendung.

Die Übertragbarkeit einer Anwendung wird gemäss ISO/IEC FCD 9126-1 bestimmt durch den Aufwand, den es benötigt, um eine Software von einer Arbeitsstation auf eine neue zu übertragen (Bevan, 1999:91). Man könnte annehmen, dass das Internet dieses Unterfangen vereinfacht hat, indem eine Applikation nicht jedes Mal neu installiert werden muss. Allerdings hat sich gezeigt, dass nicht jede Anwendung unter jedem Browser läuft, so dass auch die Übertragbarkeit von E-Learning Anwendungen gewissen Einschränkungen unterworfen ist. Sobald aber der von einer Software unterstützte Browser in der Schule vorhanden ist, wird es für Lehrpersonen mit wenig technischem Wissen einfacher, eine webbasierte Applikation auf einen neuen Computer zu übertragen.

Die Wartbarkeit bezeichnet die Fähigkeit, Software anzupassen und beinhaltet laut ISO/IEC FCD 9126-1 mehrere Komponenten (zitiert in Bevan, 1999:91):

> Modifications may include correction, improvements or adaption of the software to changes in environment, and in requirements and functional specification.

Anpassungen müssen somit auf den Ebenen des Layouts, des Inhalts und der Funktionalitäten des Programmes möglich sein. Gerade E-Learning-Anwendungen, die nicht für eine spezifische Gruppe von Schüler programmiert wurden, können häufig sinnvoller eingesetzt werden, wenn die Lehrpersonen den Lerninhalt oder das Layout den Schülern entsprechend anpassen können. Der folgende Abschnitt 3.3 beschäftigt sich ausführlicher mit den Möglichkeiten, wie Inhalt für CALL-Anwendungen aufbereitet werden kann.

9 In der Folge werden die deutschen Übersetzungen verwendet. Diese sind nicht normativ im Unterschied zu den englischen Begriffe.

Unter der Leistungsfähigkeit eines Systems versteht die ISO die Erreichung der bestmöglichen Leistung unter vorgegebenen Bedingungen (Bevan, 1999:91). Leistungsfähigkeit ist ein kritisches Thema für webbasierte CALL-Anwendungen, weil Untersuchungen gezeigt haben, dass jede für den Benutzer sichtbare zeitliche Verzögerung zwischen Browser und Server vom Lernziel ablenkt und die Anwender frustriert (Heift, 2001b:1). Leistungsfähigkeit bezieht sich aber nicht nur auf die Dauer, die zwischen Input und Lösungsausgabe einer Übung verstreicht, sondern auch auf die Zeit, die ein Anwender benötigt, um die Anwendung richtig bedienen zu lernen. Folglich sollen das GUI und die Funktionalitäten einer Anwendung selbsterklärend sein, da der User sonst wertvolle Zeit bei der Nutzung verliert (Bevan, 1999:92).

Eine CALL-Anwendung soll nicht nur effizient in der Anwendung sein, sondern auch verlässlich. Das bedeutet, dass die Applikation auch mit Fehlern, die der User macht, umgehen können muss (Bevan, 2001:540). Einerseits bezieht sich die Fehlertoleranz auf die falsche Bedienung durch den User, andererseits aber auch auf mangelhaften Eingabe durch den Benutzer. Auf diesen zweiten Typ von Verlässlichkeit geht Abschnitt 3.4 ausführlicher ein.

Die Bedienerfreundlichkeit einer Anwendung bezieht sich auf folgenden Punkt (ISO/IEC FCD 9126-1 zitiert in Bevan, 1999:91):

> the capability of the software to be understood, learned, used and liked by the user, when used under stated conditions.

Bei der Definition des GUIs muss also von Anfang an darauf geachtet werden, welche User diese Anwendung später benützen werden. Gemäss ISO beeinflussen dabei das Aussehen und die Erlernbarkeit der Schaltflächen die Akzeptanz einer CALL-Anwendung wesentlich. Heift und Nicholson weisen in ihrem Artikel «Web Delivery of Adaptive and Interactive Language Tutoring» darauf hin, dass

die Vielfalt der potenziellen Anwender einer webbasierten Anwendung eine zusätzliche Herausforderung für die Entwicklung von GUI und Schaltflächen ist (Heift, 2000:1). Sie stellen fest, dass ein System möglichst generisch sein soll, damit es von unterschiedlichen Lerngruppen verwendet werden kann. Gleichzeitig betonen sie, dass die Übungserklärungen und die Rückmeldungen auf die Lerner angepasst werden müssen. Aus diesem Grund implementieren sie für den *e-Tutor* ein umfangreiches Studentenmodell. Die Bemerkung, dass ein webbasiertes System möglichst generisch sein soll und visuelle Oberflächen und Schaltflächen nicht mehr mit Bezug auf eine bestimmte Zielgruppe gestaltet werden sollen, steht im Widerspruch zu einem user-orientierten Software-Design. Es ist wohl so, dass bei E-Learning-Anwendungen immer auch Schüler das Programm benutzen, die nicht der Zielgruppe angehören. Es macht aber keinen Sinn, die Anwendung für einen möglichst allgemeinen User zu programmieren und dabei die speziellen Bedürfnisse der eigentlichen Zielgruppe ausser Acht zu lassen. Denn damit läuft man Gefahr, dass das Aussehen des GUIs nicht mehr auf die Zielgruppe zugeschnitten und deshalb nicht ansprechend ist. Auch könnten so Schaltflächen und Anweisungen für den anvisierten Anwender nicht mehr selbsterklärend oder zumindest einfach zu erlernen sein. Wenn die Bedienung des GUIs einfach zu erlernen ist, wird Zeit und Aufwand für die Userschulung erheblich reduziert und das Vertrauen der Anwender in die Software erhöht (vgl. Bevan, 1999:92).

Genau wie für die Bedienerfreundlichkeit der User im Zentrum des Entwicklungsprozesses zu stehen hat, so soll auch die Auswahl der Funktionalitäten einer Anwendung auf den Anwender zugeschnitten werden. Für webbasierte CALL-Anwendungen sind dabei drei Punkte zentral: die Art und Weise der Rückmeldungen des Systems, der Übungstypen und der Interaktionen mit dem System durch den Anwender. Erstens muss entschieden werden, wie detailliert und

auf welche Art das System eine Rückmeldung gibt. Eine ausführliche Diskussion zu diesem Thema findet sich in Abschnitt 3.4. Zweitens muss bestimmt werden, welchen Typ Übungen die Anwendung anbietet. Schulmeister (2002:105) unterscheidet fünf mögliche Frage- und somit Aufgabentypen:

1. Ja/Nein (Richtig/Falsch)
2. Multiple Choice
3. Lückentext
4. Freie Antwort
5. Abarten dieser Grundformen

Die Aufgaben von 1 bis 3 gelten als geschlossene, 4 zählt zur Kategorie offene Übungen. Die erste Kategorie wird deshalb als geschlossen bezeichnet, weil die Reaktionsmöglichkeiten der Lernenden auf richtig oder falsch, die Auswahl aus verschiedenen Lösungsmöglichkeiten oder die Ergänzung einer Lücke begrenzt sind (Rösler, 2004:153). Immerhin garantiert diese Art von Einschränkung eine eindeutige Eingabe des Lernenden, so dass Rückmeldungen – wenigstens theoretisch – spezifisch sein können. Drittens muss bestimmt werden, ob und wie der User Übungen aus dem Übungspool auswählen kann. In Abschnitt 2.2 wurde bereits darauf hingewiesen, dass ITS Studentenmodelle haben und Übungssequenzen je nach Antworten der Lernenden angepasst werden. Es ist aber auch möglich, den Anwendern die Wahl zu lassen, welche Übungen sie wann lösen möchten. Dann muss allerdings eine Möglichkeit bestehen, die Übungen nach bestimmten Kriterien auszuwählen. Die Auswahl der Funktionalitäten sollte in das Design des GUIs einfliessen, damit die Bedienerfreundlichkeit desselben gewährleistet ist.

3.3 Wartbarkeit (maintainability) und Autorenprogramme

Wie in Abschnitt 3.1 beschrieben, bezeichnet Wartbarkeit die Fähigkeit, Software anzupassen. Technisch gesehen gibt es drei Möglichkeiten, Inhalte für CALL-Anwendungen aufzubereiten. Am einfachsten ist es, wenn die Lehrperson zusätzlichen Inhalt in ein Textfile schreiben kann und das Programm den Inhalt in ein bestimmtes Übungslayout einpasst. Ansonsten kann man sich entweder einer Autorensoftware oder einer traditionellen Programmiersprache wie Java, C++, Prolog etc. bedienen. Autorenprogramme bieten sich dann an, wenn auch Computer-Laien Lehrmaterialien mit dem Computer aufbereiten sollen.

Bereits für *PLATO* wurde ein Autorenwerkzeug (*TUTOR*) mitgeliefert, das auch Lehrenden ohne technische Ausbildung ermöglichen sollte, Lerninhalte aufzubereiten. Es gelang jedoch nicht, *TUTOR* so einfach wie möglich zu halten, weil die einzelnen Lehrpersonen, die in den Entwicklungsprozess mit einbezogen waren, unterschiedliche pädagogische Schwerpunkte setzten wollten. Sie waren sich z.B. uneinig, ob die korrekte Lösung gleich anschliessend an die Übung, erst bei erneutem Versuch oder gar nicht erscheinen sollte (Mitschian, 1999:52). Für *TUTOR* wurden am Ende alle Möglichkeiten implementiert, mit dem Resulatat, dass die Autorensprache kompliziert wurde. Den Lehrpersonen war es ohne Programmierkenntnisse nicht möglich, Lerninhalte selbst aufzubereiten.

Auch der *Bridge German Tutor* verfügte über ein Autorenwerkzeug, um neue Lerninhalte einzufügen. Allerdings waren keine Programmierkenntnisse mehr notwendig, um eine neue Übung zu generieren, da mit Templates und einer visuellen Oberfläche gearbeitet wurde (Kreyer 1995). Die Lehrpersonen benötigten einiges an Geduld und Übung, um sich in das Autorenwerkzeug einzuarbeiten. Es waren mehrere Schritte für die Erstellung einer neuen Übung nötig, weil von

der Definition der korrekten Lösung über mögliche falsche Lösungen bis hin zur Rückmeldung bei korrekten Lösungen alles von den Lehrpersonen definiert werden musste. Dadurch wurde die Einbindung von neuem Inhalt ein zeitraubendes Unterfangen.

Inzwischen gibt es eine Vielzahl von verfügbaren Autorentools wie z.B. *Mediator, Idea, Telos, Klea* oder *MMTools (*Puskás, 2005:16). Für CALL-Anwendungen wird häufig *Hot Potatoes*[10] eingesetzt, ein Autorensystem, das vom Institut Humanities Computing and Media Centre der Universität Victoria, Kanada, entwickelt wurde und dem Autorenprogramm des *Bridge German Tutor* visuell sehr stark ähnelt. *Hot Potatoes* ist in HTML und Javaskript geschrieben und besteht aus fünf frei erhältlichen Modulen[11], mit deren Hilfe webbasierte und interaktive Übungen generiert werden können. Mit dem Modul JMatch zum Beispiel können Zu- oder Anordnungsübungen erstellt werden, bei denen die Lösungmöglichkeiten in einer ungeordneten Liste Fragen einer geordneten Liste gegenübergestellt werden. Ein weiterer Übungstyp, der auf dem Modul JCloze basiert, erlaubt die Erstellung von Lückentexten, bei denen die Lernenden die Antwort selber eingeben oder aber aus einer Liste auswählen können.

Hot Potatoes arbeitet mit einer graphischen Eingabefläche und kann sehr einfach erlernt werden. Wahrscheinlich finden sich aus diesem Grund unzählige Webseiten, die auf *Hot Potatoes* basierende Übungen anbieten, wobei diese «Hotpotatoeisierung zu einer Uniformität von Übungen und Aufgaben führt, die didaktisch eigentlich unerwünscht ist» (Rösler, 2004:105). Ein weiterer Vorteil von *Hot Potatoes* (gemäss den Autoren der Software) ist die Tatsache, dass das endgültige For-

10 http://hotpot.uvic.ca/

11 Benutzer müssen sich allerdings registrieren lassen und sich verpflichten, die erstellten Übungen kostenlos auf dem Web zur Verfügung zu stellen. Ein sechstes Modul (Masher) ist gegen Gebühr erhältlich.

mat und der Stil der einzelnen Übungen einfach anzupassen ist, wenn man mit HTML und Javaskript vertraut ist. Allerdings hat es sich während einer Evaluationsphase für die vorliegende Arbeit gezeigt, dass sich das Layout der Übungen nur mit viel Aufwand anpassen lässt und die Resultate unbefriedigend sind. Um *Hot Potatoes* für e-GramD nutzbar zu machen, hätte beispielsweise der Algorithmus, der mit Hilfe von Straf- und Pluspunkten eine Prozentangabe für das Feedback ausrechnet, angepasst werden müssen. Prozentangaben können aus pädagogischen Überlegungen nicht immer verwendet werden. Der Algorithmus hätte so abgeändert werden müssen, dass er die Anzahl der korrekten Lösungen im Verhältnis zur Gesamtzahl der Fragen anzeigt. Diese Anpassung hätte in jedem einzelnen Modul vorgenommen werden müssen. Die Schwächen von Autorenprogrammen sind anhand des vorangehenden Beispieles offensichtlich: Damit Autoren ohne Programmierkenntnisse Autorenwerkzeuge benützen können, wird ein starrer Algorithmus vorgegeben, der das Format und den Stil der Übungen definiert. «Einfachheit wird folglich mit Einschränkungen in der Leistung bezahlt» (Schulmeister, 2002:103). Wenn Lehrpersonen zusätzlichen und an den Unterrichtsthemen angepassten Inhalt bereitstellen, das Layout der Übungen aber nicht ändern möchten, sind Autorenprogramme grundsätzlich die falsche Lösung. Zum einen muss die Verwendung von Autorentools erlernt werden, auch wenn graphische Oberflächen diesen Prozess vereinfachen. Viele Funktionen, die das Layout der Übungen betreffen, interessieren die Lehrenden nicht, müssen aber verstanden werden, damit Übungen überhaupt vollständig generiert werden können. Anstelle einer Autorensoftware werden daher vermehrt *Content Management System* (CMS) eingesetzt, um Lehrperson bei der Erstellung und Verwaltung von Lerninhalt zu unterstützen. Im Unterschied zu den Autorenprogrammen wird bei einem CMS Layout und Inhalt strikt getrennt. Die Inhalte werden

stattdessen mit Metadaten versehen und anschliessend in einer Datenbank abgelegt. Lerninhalte können so in kleinstmöglichste Einheiten aufgespaltet werden und für verschiedene Kurse (wieder-) verwendet werden. CMS unterstützen zudem die Organisation und Betreuung webunterstützten Lernens (Baumgartner, 2002:43).

3.4 Verlässlichkeit (reliability) bei fehlerhaftem Input

Verlässlichkeit bezieht sich auf die Fehlertoleranz von Anwendungen sowohl bei falscher Bedienung als auch bei fehlerhafter Eingabe durch den User. Letzteres stellt hohe Ansprüche an CALL-Anwendungen, gerade weil CALL-Applikationen mit fehlerhaften Eingaben umgehen können und sie verbessern müssen. Speziell ICALL-Anwendungen, die NLP-Technologien verwenden, haben Mühe mit solchen Inputs, da die meisten Parser nicht mit falschen Eingaben umgehen können. Aus diesem Grund muss ein enorm robuster Parser verwendet werden, was allerdings oft den Nachteil hat, dass dieser mehrere Lösungen liefert, aufgrund deren keine eindeutigen Rückmeldungen generiert werden können. Ansonsten muss es eine Möglichkeiten geben, fehlerhafte Inputs unterschiedlich zu behandeln und nicht in allen Fällen den Parser aufzurufen, wie dies beispielsweise beim *e-Tutor* gehandhabt wird (siehe Abschnitt 2.1.4). Der *e-Tutor* ruft dank einer zusätzlichen Analysehierarchie den Parser nicht auf, wenn z.B. Wörter in der Lösung fehlen oder zusätzliche, überflüssige Wörter eingefügt wurden. In diesen Fällen werden erst gar keine falschen Lösungen generiert (Heift 2001b:6).

Die meisten CALL-Anwendungen geben in irgendeiner Form eine Meldung – ein sogenanntes Feedback – aus, wenn die Eingabe nicht korrekt ist, obwohl die Effektivität von Fehlerkorrekturen nicht unumstritten ist (vgl. Bowerman, 1993).

Was aber versteht man unter Feedback im Unterricht überhaupt? Rösler (2004:117) gibt dazu eine nützliche Definition:

> **Feedback** ist allgemein gesehen zunächst einmal eine Rückmeldung, die jemand auf sein Verhalten erhält. Es hilft ihm, die Wirkung des eigenen Verhaltens verstehen zu lernen. Enger eingegrenzt auf den Bereich des Lernens versteht man unter **Feedback** die Informationen, die ein Lernender als Reaktion auf seine Antworten erhält und die ihm helfen sollen, die Angemessenheit seines Beitrages richtig einschätzen zu können.

Das Feedback soll einen Schüler also im Lernprozess unterstützen. Im Kontext von E-Learning-Anwendungen ist darauf zu achten, dass die Reaktionszeit, bis der Schüler eine Rückmeldung erhält, sehr kurz sein sollte. Das suggeriert Interaktivität und wirkt motivierend auf die Lernenden (Rösler, 2004:179). Die Art des Feedbacks ist je nach Applikation unterschiedlich. Es kann nach jeder Aufgabe automatisch ein Feedback ausgeben werden oder aber nur auf Wunsch des Users. Fehlerhafte Lösungen können mit «falsch» oder «richtig» kommentiert oder es kann eine genaue Fehleranalyse vorgenommen werden. Korrekte und falsche Lösungen können quantitativ ausgewertet werden und eine allgemeine Rückmeldung kann in Prozentangabe oder mit Notenzuteilung gemacht werden.

Verschiedene Autoren unterstreichen die Notwendigkeit von intelligentem Feedback (Nagata, 1993; Van der Linden, 1993; Felix, 2004; Rösler, 2004). Diesen Autoren zufolge gibt es unterschiedliche Arten von schlechtem Feedback. Es sind dies Rückmeldungen, die immer gleich lauten, egal ob ein orthografischer oder grammatischer Fehler gemacht wurde (Rösler, 2004:181). Ausserdem wird Feedback, das länger als drei Linien ist, von Lernenden nicht in ihrem Lernprozess berücksichtigt (Van der Linden, 1993:65). E-Learning-Programme, die die erste Art von ungünstigen Rückmeldungen ausgeben, lokalisieren und analysieren Fehler überhaupt nicht (Nagata, 1993:338). Im Gegensatz dazu ist die Ausgabe

von zu langen Feedbacks darauf zurück zu führen, dass keine didaktischen Überlegungen in die Generierung der Rückmeldungen eingeflossen sind.

Zur Veranschaulichung dieser Beispiele sollen hier die Rückmeldungen von drei unterschiedlichen Systemen analysiert werden: zum einen von zwei E-Learning-Systeme *Schultraining*[12] und *Lernareal*[13], welche ohne jegliche NLP-Technologie auskommen, und zum anderen vom e-*Tutor* von Trude Heift (bereits in Abschnitt 2.2 vorgestellt). Es ist zu beachten, dass *Schultraining* und *Lernareal* für den Unterricht in Deutsch als Erstsprache konzipiert wurden, der *e-Tutor* jedoch für den Zweitsprachenunterricht gedacht ist. Aus diesem Grund sind gewisse Übungen, wie das Erkennen von Wortarten, beim *e-Tutor* nicht vorhanden.

Schultraining und *Lernareal* bieten Übungen zu orthografischen und syntaktischen Phänomenen an. Interessant im Kontext der vorliegenden Arbeit sind die Übungen zu den Wortarten, die bei beiden E-Learning-Anwendungen vorhanden sind, und die Übung zum Genus von *Lernareal*.

Bei den Übungen zu den Wortarten können Schüler bei *Schultraining* in kurzen Sätzen Verben suchen und diese anklicken. Wenn die Lernenden mit der Übung fertig sind, können sie über den ok-Button ihre Lösungen überprüfen. Bei fehlerhaften Eingaben erhalten sie die Rückmeldung «Das ist teilweise richtig» (siehe Abbildung 4). Dieses Feedback ist aus zwei Gründen nicht sinnvoll: erstens wird der Fehler nicht lokalisiert, zweitens wird dem Lernenden nicht angegeben, welche Wortart «dort» ist. Wenn die Lernenden den Fehler nicht selbst finden, werden bereits beim zweiten Auswertungsversuch die Lösungen eingeblendet. Dann

12 http://www.schultraining.ch. *Schultraining* kann sowohl für die Primarstufe als auch für die Oberstufe verwendet werden.

13 http://www.lernareal.ch. *Lernareal* ist für die Oberstufe konzipiert, enthält aber z.T. Übungen, die auch für die Primarstufe verwendet werden können.

werden alle Verben, die richtig erkannt wurden, grün, falsch markierte Wörter gelb und nicht gefundene Verben rot markiert.

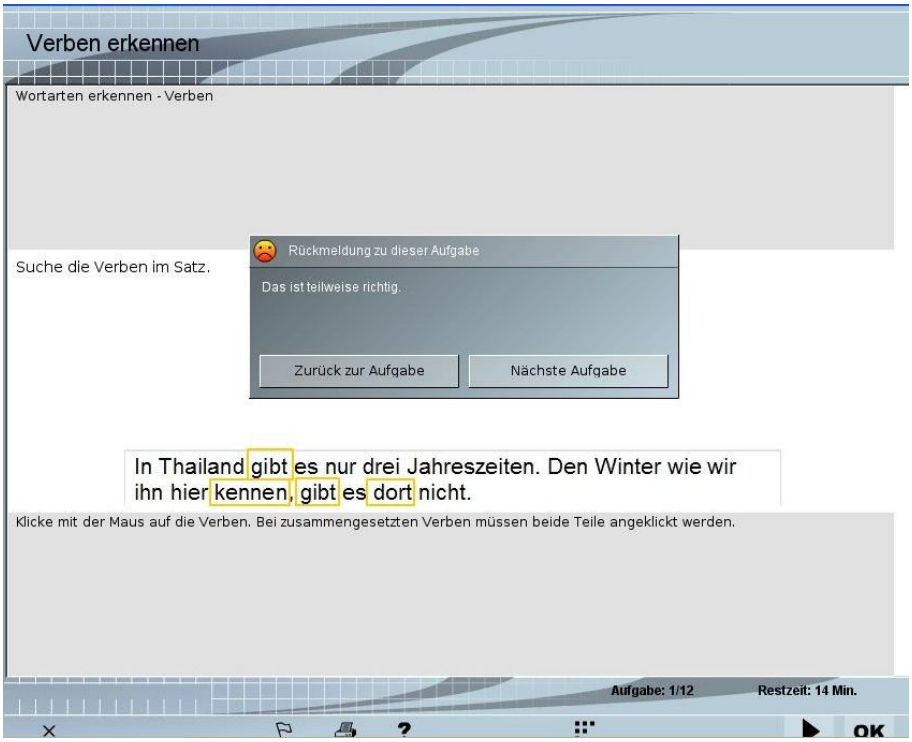


Abbildung 4: Rückmeldung Schultraining

Die Rückmeldungen von *Lernareal* unterscheiden sich bei den Worterkennungsübungen nicht wesentlich von denjenigen von *Schultraining*. Der einzige Unterschied besteht darin, dass *Lernareal* nach jedem Klick ein Feedback in einem Popup-Fenster eingeblendet wird, was bei einem längeren Text störend und nicht motivierend ist. *Lernareal* ist teilweise mit Hilfe von *Hot Potatoes* erstellt worden. Rückmeldungen über ein Dialogfeld sind typisch für Programme, die mit diesem Autorenprogramm erstellt wurden.

Mit *Lernareal* kann ausserdem das Genus von Wörtern geübt werden. Nachdem die Schüler über Drop-down-Menüs die Artikel ausgewählt haben, können sie ihre Lösungsvorschläge auswerten.

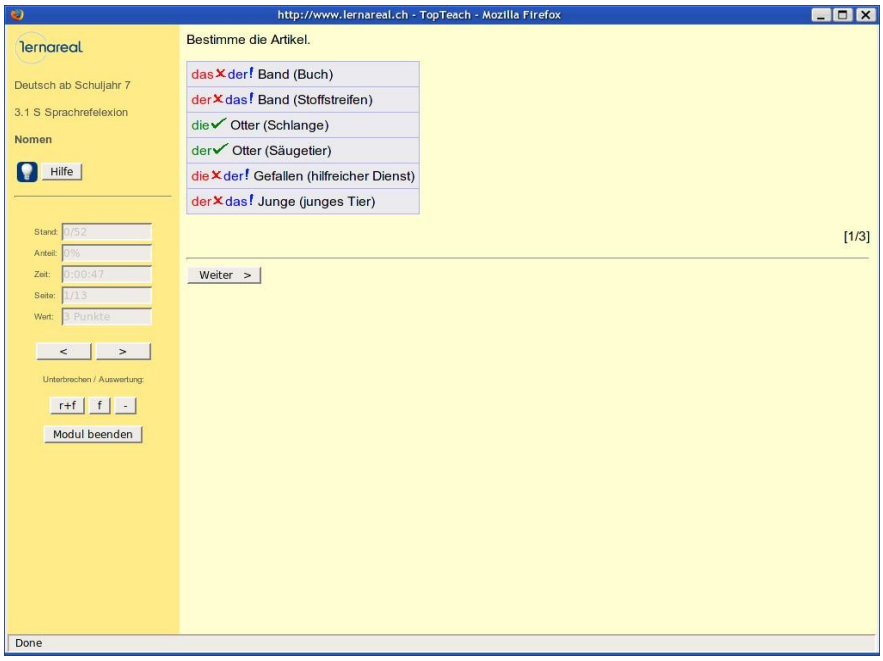


Abbildung 5: Rückmeldung Lernareal

Wiederum ist das Feedback sehr eingeschränkt: es gibt nur richtig (grüner Haken) oder falsch (rotes Kreuz). Ausserdem werden die Lösungen gleich mitgeliefert, was die Lernenden nicht zur kritischen Reflektion ihrer eigenen Eingabe anregt.

Wie in Abschnitt 2.1.4 erwähnt, ist der *e-Tutor* ein ITS-Programm, welches z.B. Lückenübungen für das Training von Kongruenzphänomenen einsetzt. Mit Hilfe eines Parsers kann eine Analyse des Satzes vorgenommen und, bei fehlerhafter Eingabe, eine entsprechende Rückmeldung geliefert werden. Deshalb erkennt der *e-Tutor* im nachfolgenden Beispiel, dass die Kongruenz zwischen dem bestimmten Artikel und dem Nomen verletzt ist. Möchten sich die Schüler die Lösung anzeigen lassen, weil sie diese selbst nicht wissen, können sie das über den «SOLVE»-Button tun.

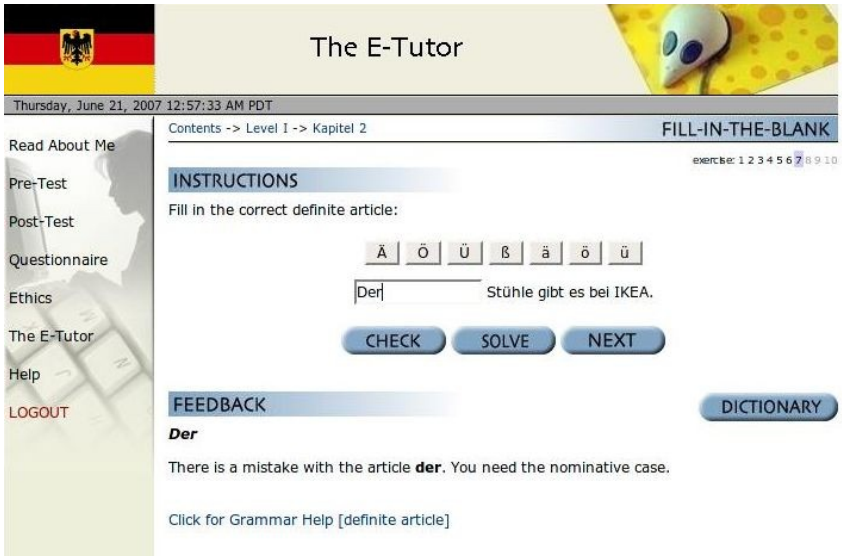


Abbildung 6: Rückmeldung e-Tutor

Betrachtet man das Feedback etwas genauer, fallen zwei Dinge auf: erstens, dass die Aufgabenstellung sehr restriktiv ist, so dass die Schüler nur einen bestimmten Artikel eintragen können. Der Parser von *e-Tutor* muss demzufolge nicht mit Lösungsvorschlägen wie «Diese Stühle gibt es bei IKEA.» umgehen können, obwohl dieser Satz auch korrekt wäre. Zweitens muss es sich beim Überlegungsfehler nicht zwingend um eine Verwechslung von Nominativ und Genitiv Plural handeln. Die Schüler könnten auch Singular und Plural des bestimmten Artikels verwechselt haben, wenn sie «der» als Lösung in die obige Übung eintragen und dann würde sie das obige Feedback wohl eher verwirren. Daher müssen die Rückmeldungen an den Lernenden aufgrund der korrekten Lösung generiert werden.

4 Grammatikunterricht Deutsch

CALL-Anwendungen konzentrieren sich auf die Vermittlung von Sprache. Eine Komponente des Sprachunterrichts ist die Ausbildung in grammatischen Phänomenen. Abschnitt 4.1 führt in die Bedeutung des Grammatikunterrichts beim Erstsprachen-Unterricht ein und beschäftigt sich mit unterschiedlichen Lerntheorien. Abschnitt 4.2 betrachtet die einzelnen grammatikalischen Kompetenzen, die im Unterricht auf der Primarstufe vermittelt werden sollen. Ausgangspunkt dieser Betrachtung ist das Lehrmittel *Grammatik,* ein Lehrmittel, das speziell auf den Deutschunterricht der Primarstufe zugeschnitten ist.

4.1 Unterrichtsformen und Lerntheorien

Um Sätze so aufbauen zu können, dass die vom Sprechenden intendierte Aussage auch vermittelt werden kann, müssen die Sprecher nicht nur über ein mentales Lexikon verfügen, sondern sie müssen auch wissen, wie sie die einzelnen Worte sinnvoll kombinieren können (Segermann, 2001:197). Beim Erwerb der L1 (Erstsprache) wird dieses grammatikalische Wissen erlernt, allerdings geschieht dies nicht bewusst. Aus diesem Grund werden in der Schule grammatikalische Phänomene gelehrt. Der Lehrplan des Kantons Aargau, der 2003 erstellt wurde, gibt für den Deutschunterricht[14] der Primarstufe vier Themenschwerpunkte vor (Aarg. Lehrplan, 2): Zuhören und Sprechen, Lesen, Texte schreiben und Sprachbetrachtung. Innerhalb der Sprachbetrachtung ist neben der Vermittlung von

14 Deutsch wird in der Schweiz als L1 unterrichtet, obwohl heute die Schüler meist ganz unterschiedlicher Muttersprache sind. Die Schwierigkeiten, die sich daraus für den Deutschunterricht ergeben, sind nicht Inhalt dieser Arbeit.

Grammatik auch das Erlernen der Rechtschreibung und das Erforschen der Sprache[15] ein Ziel. Das Erlernen von Grammatik ist somit ein Teil eines Deutschunterrichts, der alle vier Sprachkompetenzen fördern will.

Ursprünglich nahm die Vermittlung von grammatischem Wissen sehr viel mehr Platz im Unterricht ein als heute. Mit der «pragmalinguistischen Wende der 60/70 Jahre und der Neubestimmung globaler kommunikativer Lernziele» (Börner, 2001:V) erfolgte eine Neuorientierung, die eine Abkehr von der expliziten Grammatikvermittlung mit sich brachte. In den achtziger und neunziger Jahren war das Einüben von grammatischen Regeln verpönt. Erst seit Sprachenunterricht auch empirisch erforscht wird, ist belegt, dass die Vermittlung von Grammatik dennoch notwendig ist, speziell bei morphologisch reichhaltigen Sprachen wie z.B. dem Deutschen. Es stellt sich allerdings die Frage, wie Grammatik vermittelt werden kann (Börner, 2001:VI). Die Lerntheorie, die dem Unterricht zugrunde liegt, bestimmt wie Grammatik unterrichtet wird. Jedes Lehrmittel und jede Unterrichtsform basiert auf einer (oder mehreren) Lerntheorie(n), auch wenn diese selten ausdrücklich erwähnt wird. Grundsätzlich wird zwischen *behavioristischen*, *kognitiven* und *konstruktivistischen* Lernmodellen unterschieden. Oftmals wird je nach Lerninhalt und Lernziel ein anderes Modell genutzt oder verschiedene Theorien werden ergänzend eingesetzt. Das Ziel dieses Abschnittes ist es, den Lesern einen Überblick über die verschiedenen Modelle mit ihren Vor- und Nachteilen zu geben und nicht eine kritische didaktische Diskussion.

4.1.1 Behaviorismus

Der Behaviorismus ist eine Lerntheorie, die ursprünglich vom Psychologen John B. Watson entwickelt und von Burrhus F. Skinner ausgebaut wurde. Behavioris-

15 Dabei soll den Lernenden bewusst gemacht werden, wie sich eine Sprache ändert. Sie lernen mit neuen Wortschöpfungen umzugehen und wie man Nachschlagewerke benützt (Aarg. Lehrplan, 4).

mus geht von der Annahme aus, «dass Lernprozesse technisch nach dem Modell von Zweck-Mittel-Prozessen gestaltet werden können» (Moser, 1999:25). Das bedeutet, dass der Mensch beim Lernen auf Reize von aussen automatenhaft reagiert. Menschen (und somit auch Schüler) beziehen ihre Motivation aus Reaktionen der sozialen Umwelt auf ihre Handlungen, mit anderen Worten: jedes Verhalten zieht Konsequenzen nach sich. Dabei werden drei Arten von Konsequenzen unterschieden: (K+) sind positive Konsequenzen, also Lob, (K-) sind negative Konsequenzen, z.B. Strafen, und (K0) ist das Ignorieren eines bestimmten Verhaltens (Schulmeister, 2002:93). Der behavioristische Ansatz unterstreicht, dass (K-)-Reaktionen vermieden werden sollen, da Skinners Untersuchungen ergeben haben, dass jede Art von Bestrafung ein bestimmtes Fehlverhalten nur kurzfristig zum Verschwinden bringt. Unerwünschte Reaktionen der Lernenden sollen ignoriert werden, um sie dadurch zu löschen.

Als in den sechziger Jahren erste Überlegungen zum Einsatz von Computern im Unterricht angestellt wurden, erwies sich Skinners Theorie als ergiebig, weil sich das behavioristische Prinzip der «intermittierenden Verstärkung» (Schulmeister, 2002:29) und der Aufteilung des Lehrstoffs in kleinstmögliche Einheiten für die Programmierung von CALL-Software sehr gut eignet. Behavioristische Lehrmittel oder Systeme bestehen aus kleinen Einführungen in ein Thema und fragen anschliessend diese Themenblöcke mit Prüfungsfragen ab. Korrekte Antworten werden gelobt, auf falsche Lösungen wird kein Feedback gegeben. Vielmehr wird die Frage so oft wiederholt, bis die Lernenden die richtige Antwort liefern. Diese Art von Übungssequenzen wird hauptsächlich für die Vertiefung und Einübung von Faktenwissen verwendet (Drill-und Practice-Übungen). Für den Programmierer ist die Entwicklung von behavioristischer Software wesentlich einfacher als von einer kognitiven oder konstruktivistischen. Was für den Entwickler von Vorteil ist, erweist sich für den Lernenden allerdings oft als unerträglich monoton und langweilig.

4.1.2 Kognitive Ansätze

Ein Hauptpunkt der Kritik am Behaviorismus ist, dass er die inneren Prozesse der Schüler vernachlässigt. Schüler lernen nicht nur aufgrund externer Rückmeldungen, sondern auch, weil sie bei der Lösung von Problemen eigene Strategien entwickeln. Die Schüler erarbeiten sich also die Regeln für grammatische Phänomen, wie z.B. die Passivkonstruktion im Deutschen, selbständig. Daher wurden in den siebziger und achtziger Jahren vermehrt innere Abläufe, die sich beim Lernen abspielen, ins Zentrum gestellt und kognitive Lernstrategien entwickelt.

Allerdings erfordern kognitive Aufgabenstellungen einiges an Vorbereitung seitens der Lehrpersonen (Kerres, 2001:67): So müssen Unterrichtsform und -situation sorgfältig ausgewählt werden, damit ein vorab bestimmtes Unterrichtsziel auch tatsächlich erreicht werden kann. Ausserdem müssen auf irgendeine Art die Lernzustände überprüft werden. Im Gegensatz zur unmittelbaren Rückmeldung beim Behaviorismus steht beim Kognitivismus ein informatives Feedback im Vordergrund (Schulmeister, 2002:109). Die Lehrpersonen stehen während des Unterrichts als Tutoren zur Verfügung und fördern die Regelfindung durch intelligentes Feedback. Mit Hilfe der Rückmeldungen sollen die Lernenden über mögliche Ursachen ihrer Probleme informiert werden und dann anschliessend eigenständig das weitere Lernen steuern. Daher unterstützten kognitivistische Lernmaterialien eine grössere Selbstbestimmung der Schüler bei der Wahl des Schwierigkeitsgrades, der Problemstellung und des Tempos der Lösungsfindung. Es erstaunt deshalb nicht, dass auch von E-Learning-Anwendungen mehr Adaptivität verlangt wird, wenn sie dem kognitiven Lernmodell entsprechen sollen. Kognitive Ansätze wurden vor allem in den ITS implementiert. Der Computer übernimmt dabei die Rolle des Tutors. Die Systeme gelten als intelligent, weil sie die Lerninhalte flexibel arrangieren und adaptiv auf den Wissenstand und die kognitive Leistung des Lernenden eingehen können (Schulmeister, 2002:187). Es ist allerdings zu

beachten, dass die Konzeption und die technische Implementation solcher E-Learning-Anwendungen sehr aufwändig sind, sie häufig nicht über das Stadium eines Prototypen herauskommen und die meisten Ansätze weit davon entfernt sind, kognitive Merkmale der Informationsverarbeitung adäquat zu modellieren.

4.1.3 Konstruktivistische Ansätze

Kognitive Ansätze unterstützen hauptsächlich die Entwicklung von Suchstrategien zu vorgegebenen Problemstellungen. Dabei bleibt die Kontrolle über die zu lösenden Probleme bei den Lehrpersonen. Ende der achtziger Jahre hielten Konstruktivisten den Kognitivisten entgegen, dass Schüler auch adäquat auf ihnen unbekannte Problemstellungen in Alltagssituationen reagieren können. Vertreter des Konstruktivismus[16] gehen daher davon aus, dass Schüler eigene Wissenswelten konstruieren (Moser, 1999:26), um für sie wichtige Probleme lösen zu können. Im Unterricht wird Konstruktivismus häufig mit situiertem Lernen in Verbindung gebracht: Damit Schüler relevante Probleme generieren können, wird der Lerninhalt in eine praxisrelevante Situation eingebettet. Schüler können im Idealfall selbst Problemsituationen generieren. Lehrpersonen stehen als Trainer zur Verfügung, die – je nach Situation – einen Prozess erklären und vorzeigen, die Lernenden in die Problematik einführen, sich dann aber möglichst bald ausblenden oder die Schüler während des Prozesses begleiten. Schliesslich wird von den Lehrpersonen erwartet, dass sie zusammen mit den Schülern den Prozess reflektieren. Die technischen Entwicklungen in den neunziger Jahren – wie die Verbreitung von Internet oder die Entwicklung von multicodalen Formaten – unterstützten die Entwicklung von konstruktivistischen E-Learning-Anwendungen, wie z.B. *LernNetz Deutsch*[17] (Donath 1998). Die Aufgaben in *LernNetz Deutsch* sind situations-

16 Es sei darauf hingewiesen, dass Konstruktivismus nicht ausschliesslich eine Lerntheorie bezeichnet, sondern auch als Wissenschafts- und Erkenntnistheorie oder als Menschenbild definiert werden kann.

17 http://lerndeutsch.com/

bezogen und nutzen die Vorteile der verschiedenen Medien. So werden die Pronomen über das Märchen «Rumpelstilzchen» vermittelt, und die Lernenden können während dem Lesen Wörter nachschlagen, die sie nicht kennen. Anschliessend stehen verschiedene Aktivitäten zur Auswahl: es können Fragen zum Text beantwortet, eine kurze Einführung in die Pronomen gelesen und anschliessend Übungen dazu gelöst oder weiterführende Informationen zu den Gebrüder Grimm nachgelesen werden. *LernNetz Deutsch* gibt den Lernenden also die Möglichkeit, in vielen verschiedenen Informationsquellen zu recherchieren sowie Aufgaben mit Unterstützung diverser Werkzeuge zu lösen. Die Autoren von *LernNetz Deutsch* weisen klar darauf hin, dass die Lernumgebung ausschliesslich für fortgeschrittene Lernende empfehlenswert ist. Dies ist – neben der technisch aufwändigen Entwicklung konstruktivistischer Anwendungen – auch der hauptsächliche Nachteil von konstruktivistischen Ansätzen.

4.2 Grammatikunterricht auf Primarstufe

In der Regel beschäftigt sich der Grammatikunterricht mit der Vermittlung von Phonetik[18], Morphologie und Syntax (Flückiger, 2004:28). In der Grundschule wird Grammatik unter anderem mit dem Lehrmittel *Grammatik* (Hochstrasser, 2006) unterrichtet, welches einige eindeutige Merkmale des behavioristischen Ansatzes aufweist. Der Aufbau der einzelnen Themenblöcke folgt immer dem gleichen Muster: vorab werden einige Anhaltspunkte zum Inhalt des Lernziels präsentiert, dann werden diese Strukturen mit Hilfe von Übungen abgefragt und anschliessend von der Lehrperson korrigiert. Allerdings werden – in Anlehnung an die kognitiven Vorstellungen – falsche Lösungen nicht nur korrigiert, sondern mit zusätzlichem Feedback versehen, damit die Lernen-

18 Allerdings wird im Grammatikunterricht der L1 auf der Ebene der Phonetik keine Aussprache geübt, weil dieses Wissen bereits unbewusst angeeignet wurde, sondern das Erkennen von Silben und Betonungen. Dieses Wissen wird benötigt, um Orthografie zu vermitteln. Dieser Bereich ist aber nicht Thema der vorliegenden Arbeit und wird daher nicht weiterverfolgt.

den die Fehler selbständig verbessern können. Die Texte und Übungen von *Grammatik* sind in Abstimmung mit dem zu vermittelnden Grammatikpensum geschrieben und beinhalten Standardtexte zu Zirkus oder Schifffahrten. Sie sind daher nicht situationsbezogen und für die Schüler nicht praxisrelevant. Rösler (2004:79) hingegen fordert, dass Lerninhalte «in Kontexte des Alltags eingebettet werden», damit sie für die Lernenden interessanter sind.

Das Lehrmittel *Grammatik* führt Grammatik auf Primarstufe ein, ohne dass eindeutig zwischen den linguistischen Bereichen wie Syntax oder Semantik unterschieden wird. Dies wohl deshalb, weil die klassische Trennung in die einzelnen linguistischen Bereiche in der Realität kaum aufrechterhalten werden kann. So wird die Zuordnung zu den Wortklassen z.B. traditionell als Aufgabe der Morphologie betrachtet, weil morphologische Kriterien wie die Flexionsmerkmale zur Zuordnung herbeigezogen werden können. Es wird allerdings schnell klar, dass diese Zuordnung auch über die Semantik stattfinden kann. Ein Lernender kann sich nämlich auch auf die Tatsache berufen, dass «Autos» oder «Laufschuhe» Objekte in der Welt sind, und die Einordnung in die Wortklasse der Nomen aufgrund dieser Beobachtung vornehmen. Bei abstrakten Nomen ist dieser Weg allerdings schwieriger, hier können aber syntaktische Patterns (Begleiter-Nomen) eine Hilfestellung bieten. Wie dieses Beispiel zeigt, ist es für pädagogische Zwecke nicht sinnvoll, zwischen linguistischen Betrachtungsweisen zu unterscheiden. Vielmehr geht es darum, das implizite Wissen über die Veränderbarkeit von Wörtern, die bestehenden Abhängigkeiten in einem Satz oder semantische Organisationsprinzipien der Wörter explizit zu machen. Es erstaunt daher nicht, dass das Lehrmittel *Grammatik* nach den einzelnen Lernzielen geordnet ist und die Arbeitsblätter dieser Aufteilung folgen (Hochstrasser 2006)[19]. Nachfolgend werden die einzelnen, für die Ent-

19 Die Auflistung der Lernziele und eine Auswahl der Arbeitsblätter sind im Anhang A aufgeführt.

wicklung von *e-GramD* relevanten, Lernziele nach linguistischen Kriterien aufgegliedert vorgestellt.

4.2.1 Morphologische Kompetenzen

Deutsch ist eine flektierende Sprache und die verschiedenen Flexionsformen unterscheiden sich durch grammatische Merkmale voneinander. Die nachfolgende Tabelle 1 stellt die einzelnen Flexionsmerkmale des Deutschen den Wortarten gegenüber.

Merkmalklasse	*einzelne Merkmale*	*Wortart*
Zahl (Numerus)	Singular, Plural	Nomen, Adjektive, Pronomen
Geschlecht (Genus)	männlich, weiblich, sächlich	Nomen, Adjektive, Pronomen
Fall (Kasus)	Nominativ, Genitiv, Dativ, Akkusativ	Nomen, Adjektive, Pronomen
Person	1. Person, 2. Person, 3. Person	Verben, Pronomen
Zeit (Tempus)	Präsens, Perfekt, Präteritum, Plusquamperfekt, Futur I, Futur II	Verben
Aussageweise (Modus)	Indikativ, Imperativ, Konjunktiv I, Konjunktiv II	Verben
Handlungsrichtung	Aktiv, Passiv	Verben
Komparation	Positiv, Komparativ, Superlativ	Adjektive

Tabelle 1: Flexionsklassen des Deutschen (basierend auf Flückiger, 2004:30)

Die Veränderungsmerkmale, die bei den Verben zu finden sind, werden unter dem Begriff *Konjugation* zusammengefasst, alle anderen unter dem Begriff *Deklination*. Die Übungsinhalte von *e-GramD* konzentrieren sich vorläufig auf Deklinationsphänomene und daher wird alles, was sich auf Verben bezieht, im Weiteren ausgeklammert.

Die morphologischen Lernziele in der vierten Klasse decken vier von fünf Wortarten ab, nämlich Nomen, Verben, Adjektive und Pronomen. Die nicht flektierbaren Partikel[20] werden erst in der fünften Klasse eingeführt. Nebst dem Erkennen der einzelnen

20 Zu den Partikeln werden Präpositionen, Adverbien, Konjunktionen und Interjektionen gezählt.

Wortarten steht vorerst die Vermittlung von Numerus und Genus auf dem Lehrplan, in der fünften Klasse wird schliesslich auch Wissen über den Kasus vermittelt. Beim Numerus wird den Lernenden von Anfang an vermittelt, dass nicht alle Nomen bei der Pluralbildung verändert werden. Bei denjenigen, die eine Modifikation erfahren, wird gezeigt, dass die Flexion sowohl durch Anhängen von Endungen (Suffixe) als auch durch eine Veränderung im Wortinnern (Umlaute) stattfinden kann, auch wenn auf theoretische Erklärungen verzichtet wird. Als Regeln erhalten die Lernenden folgende drei Anhaltspunkte (Hochstrasser, 2006:19):

> Bei den meisten Nomen werden im Plural ein oder mehrere Buchstaben angehängt.
>
> Beispiele: der Hund – die Hund**e**, das Kind – die Kind**er**, das Velo – die Velo**s**
>
> Bei einigen Nomen tönen der Singular und der Plural genau gleich.
>
> Beispiele: der Spiegel – die Spiegel, der Teller – die Teller
>
> Bei einigen Pluralformen entstehen Umlaute.
>
> Beispiele: das Haus – die H**äu**ser, das Schloss – die Schl**ö**sser

Die Zuordnung zu den Deklinationsreihen wird im Grammatikunterricht beim Erlernen von Deutsch als Zweitsprache normalerweise mit Wissen über die Deklinationsreihen[21] vermittelt. Grundschüler erhalten aber einzig eine Auflistung aller möglichen Pluralsuffixe. Deutsch wird in der Schweiz als L1 unterrichtet in der Annahme, dass sich die Kinder auf ihre muttersprachlichen Erfahrungen im Alltag stützen können und so die Zuordnung zu den einzelnen Flexionsmerkmalen des Numerus (oder auch des Genus) ohne weitere Schwierigkeiten vornehmen können.

21 Ich beziehe mich hier auf die drei Deklinationsreihen Nulldeklination, s-Deklination und n-Deklination (vgl. Flückiger, 2004:79) .

Ein weiteres morphosyntaktisches Lernziel der vierten Klasse ist das Wissen über Kompositionsphänomene. Dabei steht bei den Nomen die Bildung von Komposita im Vordergrund. Hierzu wird die einfache Regel aufgestellt, dass Komposita entweder aus mehreren Nomen oder einem Verb und einem Nomen bestehen können (Hochstrasser, 2006:23). Das ist eine Vereinfachung, denn Komposita können auch mit Adjektiven (Grünschnabel) oder mit Pronomen (Selbstwertgefühl) gebildet werden. Den Autorinnen von *Grammatik* geht es darum, die Aufmerksamkeit auf bestimmte Phänomene zu lenken, und nicht darum, komplizierte oder vollständige Entscheidungsraster zu vermitteln.

4.2.2 Syntaktische Kompetenzen

Wie bereits erwähnt, wird ab der zweiten Klasse die Erkennung der einzelnen Wortarten geübt, anfänglich nur der offenen Klassen (Nomen, Verben, Adjektive[22]); bis in der sechsten Klasse sind auch die geschlossenen Klassen (Pronomen und Partikel) vollständig erarbeitet. Dabei steht von Anfang an die Betrachtung ganzer Wortgruppen im Zentrum, d.h. es werden so genannte *Patterns* vermittelt: Im Falle der Nominalgruppe besteht das Pattern aus Begleiter und Nomen, und die Zuordnung der Nomen zu den verschiedenen Genera wird mit Hilfe der Begleiter vorgenommen. Segermann erklärt, dass Schüler diese Patterns ohnehin berücksichtigen, wenn sie einen Gedanken in Worte fassen (2001:197). Durch die Konzentration auf Patterns soll eine künstliche Entkoppelung von Syntax und Morphologie vermieden und die Verwendung der einzelnen Wortformen aus dem Satzzusammenhang erklärt werden (Flückiger, 2004:140). Durch die aktive Beschäftigung der Lernenden mit den eigenen impliziten Mustern werden diese explizit gemacht. Bis Ende der vierten Klasse werden so bestimmter und unbe-

22 Interessanterweise werden die Adjektive gelernt, ohne die syntaktischen Funktionen zu berücksichtigen. So wird «schön» von «Thomas singt schön.» als Adjektiv markiert, obwohl es eine adverbiale Funktion innehat.

stimmter Artikel, Possessivpronomen, Demonstrativpronomen und bestimmte Zahlpronomen eingeführt, freilich ohne diese detaillierte linguistische Klassierung.

Ein weiteres syntaktisches Lernziel der vierten Klasse besteht darin, Sätze zu erweitern, zu kürzen oder umzustellen. Dabei werden wieder ganze Wortgruppen betrachtet, allerdings werden sie weder funktional als Satzglied noch als Phrasenkonstituente beschrieben, wie das in der Linguistik üblich ist. In der folgenden Übung sollen Nominal-, Verbal- und Präpositionalphrase in zwei korrekte, aber unterschiedliche Sätze umformuliert werden (Hochstrasser, 2006:93). Dabei grenzen die Rahmen die einzelnen Phrasen ab.

Ich	hole	die Zeitung	aus dem Briefkasten

In der fünften und sechsten Klasse lernen die Kinder, die Wortgruppen selbst zu bestimmen, indem sie sich aktiv die Technik der Verschiebungsprobe aneignen. Die Satzlehre wird für meine Arbeit nicht weiterverfolgt, da dies den Rahmen des Entwicklungsaufwands für *e-GramD* sprengen würde.

4.2.3 Semantische Kompetenzen

Wie bereits in der Einleitung von Abschnitt 4.2 beschrieben, kann die Zuordnung von Nomen auch über semantische Kriterien erfolgen. Den Lernenden wird sowohl über die Morphologie und die Syntax als auch über die Semantik die Erkennung der Nomen erklärt, weil sehr viele Schüler Mühe damit bekunden und dies einen Einfluss auf die Rechtschreibung dieser Wortklasse hat. Das Kleinschreiben von Nomen wird oft als Orthografiefehler klassifiziert, was allerdings nach den Erfahrungen der für dieses Projekt hinzugezogene Grundstufenlehrerin nur bedingt stimmt. Schüler schreiben ein Nomen nur dann klein, wenn sie das Wort nicht der entsprechenden Wortklasse zuordnen können. Daher ist die Gross- und Kleinschreibung als grammatikalische Kompetenz zu betrachten.

Ausserdem beschäftigen sich die Lernenden der vierten Klasse mit semantischen Beziehungen zwischen Nomen. Dabei handelt es sich ausschliesslich um Übungen zum Phänomen der Hyponymie/Hyperonymie, also der Zuordnung von Unterbegriffen zu Oberbegriffen. Die Tatsache, dass «Autos» oder «Schiffe» auch «Fahrzeuge» sind, wissen die Kinder intuitiv, der Grammatikunterricht soll das Wissen explizit machen. Dieses Lernziel wird für die vorliegende Arbeit nicht weiterverfolgt.

Die vorliegende Arbeit konzentriert sich auf die Implementation von morphosyntaktischen Phänomenen, die in der vierten Klasse vermittelt werden. *E-GramD* bietet Übungen mit Hilfe derer das Erkennen der einzelnen Wortarten geübt werden kann. Ausserdem kann die Erkennung der verschiedenen Flexionsendungen trainiert werden. Schliesslich können Flexionen auch aktiv geübt werden, da Lückentexte implementiert wurden.

5 ICALL- und CALL-Anwendungen im Vergleich

Die vorangehenden Kapitel haben gezeigt, dass für die Entwicklung von E-Learing-

Anwendungen unterschiedliche Kriterien und Massstäbe im Vordergrund stehen können. Sowohl technische wie auch medienpädagogische Aspekte haben einen Einfluss auf das Endresultat. Ziel dieses Kapitel ist es einerseits, einen Überblick über den Status quo von E-Learning-Anwendungen zu geben. Andererseits werden die Vor- und Nachteile von traditionellen CALL- respektive ICALL-Applikationen herausgeschält. In Abschnitt 5.1 wird dazu ein Kriterienkatalog erarbeitet, der technische, didaktische und medienpädagogische Aspekte berücksichtigt. Abschnitt 5.2 gibt einen Überblick über zwölf Applikationen, die mit Hilfe der Kriterien charakterisiert wurden. Die Beschreibung und Diskussion der qualitativen Analyse stehen im Zentrum von Abschnitt 5.3.

5.1 Kriterienkatalog

Es gibt bereits verschiedene Kriterienkataloge, mit denen E-Learning-Anwendungen untersucht werden können: Levy (2006) zum Beispiel erarbeitet einen Kriterienkatalog, der allgemein für WBT-Anwendungen (web-based Training) gilt, unabhängig vom Lerninhalt. Baumgartner und Maier-Häfele (2002) sowie Rösler (2004) konzentrieren sich auf webbasierte CALL-Anwendungen und stellen medienpädagogische Gesichtspunkte ins Zentrum. Gamper und Knapp (2003) hingegen analysieren ausschliesslich ICALL-Anwendungen und konzentrieren sich auf die verwendeten Technologien der künstlichen Intelligenz. Die unter-

schiedlichen Sichtweisen ergeben sich aus den verschiedenartigen Zielen der Evaluationen. Das hier erarbeitete Raster soll sich nicht auf ein bestimmtes Merkmal konzentrieren. Eine Anwendung darf nicht nur technisch hochstehend sein, sie muss vor allem auch in den pädagogischen Alltag integriert werden können. Daher werden technische, didaktische und medienpädagogische Aspekte berücksichtigt. Hierzu fliesst die Diskussion zu den Usability-Standards (vgl. Kapitel 3) in die Auswahl der Kriterien für den Katalog mit ein. Tabelle 2 bietet einen Überblick diese Kriterien und wird für die weitere Arbeit als Referenzrahmen benutzt.

Schwerpunkt		*Kriterium*	*Art*
Kontext		Einsatz	technisch
		Lernsprache	didaktisch
		L1/L2	didaktisch
		sprachliche Kompetenz	didaktisch
		Unterrichtsstufe	didaktisch
		Unterrichtstheorie	didaktisch
Anwendungsqualität	Übertragbarkeit	Art der Anwendung	technisch
	Wartbarkeit	File, Autorensoftware, Programmiersprache	technisch
	Leistung	Verarbeitungszeit	technisch
	Verlässlichkeit	Reaktion bei Eingabefehler	technisch
	Bedienerfreundlichkeit	Bedienerfreundlichkeit Anweisungen	medienpädagogisch
		Bedienerfreundlichkeit Schaltflächen	medienpädagogisch
		Bedienerfreundlichkeit Hilfe	medienpädagogisch
	Funktionalitäten	Feedback	medienpädagogisch
		Übungstypen	medienpädagogisch
		Interaktion	medienpädagogisch

Tabelle 2: Kriterienkatalog CALL Anwendungen

Es gibt zwei Hauptgruppen: Kriterien, die den Kontext, in dem eine Software eingesetzt wird, beschreiben, und solche, welche die Anwendungsqualität (Übertragbarkeit,

Leistungsfähigkeit, Wartbarkeit, Bedienerfreundlichkeit und Funktionalitäten) der Applikation definieren. Diesen zwei Themenschwerpunkten können weitere Kriterien zugeordnet werden. Bei jedem Kriterium wird jeweils unterschieden, ob es technischer, didaktischer oder medienpädagogischer Natur ist.

Der Kontext, indem eine Software eingesetzt wird, kann zunächst über den Einsatz der Applikation bestimmt werden. Systeme können kommerziell eingesetzt werden, frei verfügbar oder als Prototyp konzipiert sein. Prototypen müssen nicht die gleichen Anforderungen wie ein kommerziell vertriebenes System erfüllen. Sie dienen z.B. dazu, die technischen Möglichkeiten eines bestimmten Ansatzes der künstlichen Intelligenz auszutesten. Teilweise werden sie von Anwendern evaluiert, ein längerer Einsatz im Unterricht steht für die Entwickler aber nicht im Vordergrund, so dass pädagogische Aspekte in den Hintergrund treten. Für ein System, das in den Unterricht integriert werden soll, muss der didaktische Kontext hingegen klar definiert werden. Die Lernsprache bestimmt grundsätzlich, welche Sprache mit der Applikation erlernt werden soll. Ist dieser Punkt eingegrenzt, muss erfasst werden, ob die Anwendung in erster Linie für den L1- oder den L2-Unterricht gedacht ist. Eine L2-Software kann Menüfunktionen, Hilfestellungen und Feedback in der L1 der Lernenden oder in der L2 haben, je nachdem auf welcher Lerntheorie die Applikation basiert. Die ausgewählte Lerntheorie bestimmt ausserdem, welche Art von Rückmeldungen ausgegeben wird. Welche sprachlichen Fähigkeiten mit dem System trainiert werden können, muss als nächstes bestimmt werden. Sehr viele CALL-Applikationen bieten inzwischen eine Gesamtlösung an, so dass sowohl Lese-, Hör- und Grammatikübungen von den Schüler gelöst werden können. Es gibt aber auch verschiedene Anwendungen, die sich auf die Vermittlung einer bestimmten sprachlichen Kompetenz konzentrieren. Die meisten Systeme sind für eine bestimmte Benutzergruppe konzipiert worden, auf die das Layout, die

Art der Rückmeldungen und der Hilfestellungen angepasst wurde. Eine Webseite für Erwachsene spricht Zwölfjährige nicht an und umgekehrt.

Erst wenn der Einsatz einer Anwendung bestimmt ist, können Kriterien der Anwendungsqualität betrachtet werden. Dabei gilt es zu definieren, wer die Applikation von einem System auf das nächste überträgt. Gleichzeitig muss geklärt werden, ob es Einschränkungen betreffend Betriebssystemen oder Browsern in der anvisierten Unterrichtsstufe gibt, auf denen die Anwendung später laufen soll. Wie bereits erwähnt, bietet sich eine webbasierte Anwendung an, wenn Lehrpersonen keine technische Unterstützung in Anspruch nehmen können, eine Internetverbindung aber zur Verfügung steht. Bei einer Neuinstallation muss in diesem Fall lediglich ein entsprechender Browser zur Verfügung stehen. Es muss weiter geklärt werden, ob Lehrpersonen zusätzlichen Inhalt für die Anwendung generieren wollen. Ist dies der Fall, muss die Anwendung den Lehrern auf irgendeine Art Unterstützung bieten. Können sie zusätzliche Texte einfach einlesen und daraus Übungen generieren, oder können sie mit einer Autorensoftware oder einer Programmiersprache Inhalt und Layout anpassen? Wie schnell das System eine Eingabe verarbeitet, ist ein weiterer Entscheidungspunkt. Da jedes CALL-System mit Falscheingaben umgehen können muss, soll auch dieser Punkt genauer betrachtet werden. Dabei wird berücksichtigt, ob ein System Methoden der künstlichen Intelligenz einsetzt. Die Art der Rückmeldungen wird nicht in diesem Zusammenhang sondern bei den Funktionalitäten der Anwendung diskutiert. Zum Thema Bedienerfreundlichkeit werden gleich drei Aspekte berücksichtigt: erstens, wo und wie das System dem Lernenden Anweisung zur Übung gibt. Zweitens wird betrachtet, ob es verschiedene Schaltflächen gibt für die Analyse des Inputs und das Anzeigen der korrekten Lösung und wie verständlich die Beschriftung und Anordnung derselben ist. Drittens wird analysiert, ob es eine Hilfefunktion gibt, die den Lernenden unterstützt, wenn er bei einer Aufgabe nicht mehr weiterkommt. Bei den Funktionalitäten einer

Anwendungen gilt das Augenmerk vor allem den Rückmeldungen, den Übungstypen und der Interaktion mit der graphischen Oberfläche, d.h. ob und wie die Schüler Übungen auswählen können.

5.2 Datensammlung und -erhebung

Gegenstand der Untersuchung waren zwölf unterschiedliche CALL-Systeme. Die Ausschlusskriterien bildeten die Lernsprache und die zu trainierende Fertigkeit. Alle analysierten Anwendungen können somit im Deutschunterricht für das Erlernen von Grammatikkompetenzen eingesetzt werden. Ausgewählt wurden Applikationen, die entweder in wissenschaftlichen Artikeln im Detail beschrieben sind oder von Lehrpersonen ausdrücklich empfohlen wurden und die in den letzten 20 Jahren entwickelt wurden. Die verschiedenen Systeme sind allerdings sehr unterschiedlich, abhängig davon, wie Grammatik vermittelt werden soll. Alle Anwendungen stellen auf irgendeine Art und Weise Trainingsübungen zur Verfügung, einige liefern zudem Hintergrundinformationen zum Grammatikpensum (z.B. *Eldit IV*).

Tabelle 3 bietet eine Übersicht über die untersuchten Anwendungen. Die mit einem * gekennzeichneten Anwendungen konnten getestet werden. Bei Komplettlösungen wie *TellMeMore* oder dem *e-Tutor*, mit denen u.a. auch Hör- oder Leseverständnis trainiert werden können, wurde nur das Grammatikmodul betrachtet. Die Anwendungen wurden nicht auf die Aspekte der Bedienerfreundlichkeit untersucht, weil diese Evaluation sehr subjektiv ist und sie von einer grösseren Zahl von Anwendern der Zielgruppe durchgeführt werden muss, damit die Resultate aussagekräftig sind. Von den zwölf CALL-Anwendungen sind sechs ICALL-Applikationen, die anderen sechs sind traditionelle Systeme. Vier der untersuchten ICALL-Systeme verwenden NLP-Tools: der *e-Tutor*, der *Bridge German Tutor*, *Robi* und *Lice*. *Lice*, *Herr Kommissar*, der *Bridge German Tutor* und der *e-Tutor* sind ITS, *Eldit IV* ein sogenanntes adaptives System (Gam-

per, 2002:330). Acht der evaluierten Systeme werden in wissenschaftlicher Literatur erwähnt (Donath, 1998; Gamper, 2002; Rösler, 2003), vier wurden von Lehrpersonen der Primarstufe empfohlen. Es ist bemerkenswert, dass die Lehrpersonen auf keine einzige ICALL-Applikation verwiesen haben.

Name	*Art*	*ausgewählt anhand von*	*seit*
Bridge German Tutor	ICALL	Literatur	1993
e-Tutor * http://www.e-tutor.org	ICALL	Literatur	2000
Eldit IV * http://dev.eurac.edu:8081/MakeEldit1/dicturl	ICALL	Literatur	2002
German Steps * http://www.bbc.co.uk/languages/german/lj/	CALL	Lehrperson	2002
Herr Kommissar	ICALL	Literatur	1993
Interdeutsch * http://www.interdeutsch.de/eng/index.html	CALL	Literatur	1998
Lernareal * http://www.lernareal.ch/	CALL	Lehrperson	2006
LernNetz Deutsch * http://lerndeutsch.com/	CALL	Literatur	1996
Lice http://www.cet.sunderland.ac.uk/cbowww/SOFTWARE/LICE/	ICALL	Literatur	1993
Robi nur in Literatur berschieben	ICALL	Literatur	2005
Schultraining * http://www.schultraining.ch/	CALL	Lehrperson	2005
TellMeMore http://h16.e-tmm.com/portal/modportal.axrq	CALL	Lehrperson	1987

Tabelle 3: Überblick evaluierte Anwendungen

5.3 Diskussion der erhobenen Daten

Die Beschreibung und Diskussion der Daten ist in zwei Teile gegliedert: Abschnitt 5.3.1 befasst sich mit dem Kontext der Anwendung und Abschnitt 5.3.2 mit den Punkten der Anwendungsqualität.

5.3.1 Untersuchte Anwendungen im Kontext

Es wurden sechs Aspekte untersucht, die bestimmen, in welchem Kontext eine Anwedung eingebettet ist: Einsatz, Lernsprache, L1 oder L2, sprachliche Kompetenz, Unterrichtsstufe und -theorie (vgl. Tabelle 2). Sowohl Lernsprache wie auch sprachliche Kompetenz waren Ausschlusskriterien, nach denen die zwölf Anwendungen ausgewählt wurden. Daher werden diese zwei Aspekte in diesem Abschnitt nicht mehr erwähnt.

Tabelle 4 listet die kontextbestimmenden Merkmale der sechs traditionellen CALL-Applikationen auf, die untersucht wurden. Bei den traditionellen CALL-Systemen finden sich sowohl frei verfügbare Anwendungen wie solche, die kommerziell vertrieben werden: *Lernareal*, *LernNetz Deutsch* und *German Steps* sind frei verfügbar, *Schultraining*, *Interdeutsch* und *TellMeMore* werden kommerziell vermarktet.

traditionelle CALL						
Kriterium	*Lernareal*	*LernNetz Deutsch*	*Schultraining*	*Interdeutsch*	*TellMeMore*	*German Steps*
Einsatz	frei verfügbar	frei verfügbar	kommerziell	kommerziell	kommerziell	frei verfügbar
L1/L2	L1	L2	L1	L2	L2	L2
Unterrichtsstufe	Oberstufe	Oberstufe	Primarstufe	Erwachsene	Erwachsene	Oberstufe
Unterrichtstheorie	Behaviorismus	Konstruktivismus	Behaviorismus	Behaviorismus	Konstruktivismus	Konstrukvismus

Tabelle 4: traditionelle CALL-Anwendungen im Kontext

Unabhängig vom Einsatz zeigt sich, dass die Entwickler von traditionellen CALL-Anwendungen jeweils eine bestimmte Zielgruppe im Fokus hatten. Um deren Ansprüche zu erfüllen, werden die restlichen drei Kriterien sehr unterschiedlich gehandhabt.

Lernareal und *Schultraining* sind Anwendungen, die ausdrücklich für den Erstsprachenunterricht konzipiert wurden. So sind Anweisungen, Hilfestellungen und Rückmeldungen auf Deutsch. Obwohl *LernNetz Deutsch* für den L2-Unterricht gedacht ist, navigieren die Schüler auch hier über deutsche Menüs und lesen deutsche Anweisungen. Das hängt mit zwei Entscheidungen zusammen: erstens basiert *LernNetz Deutsch*

auf einem einsprachigen Ansatz, d.h. der Unterricht soll ausschliesslich in der Fremdsprache erfolgen. Zweitens ist die Lernumgebung ausschliesslich für fortgeschrittene Lernende gedacht (vgl. Abschnitt 4.1), die deutsche Aufforderungen bereits verstehen können. *InterDeutsch*, *TellMeMore* und *German Steps* sind für den L2-Unterricht konzipiert, unterstützen die Lernenden aber mit Hilfestellungen entweder in Englisch (*German Steps*, *InterDeutsch*) oder in ihrer Muttersprache (*TellMeMore*).

Schultraining ist die einzige Anwendung, die Übungen für die Primarstufe anbietet. *LernNetz Deutsch*, *Lernareal* und *German Steps* sind für die Oberstufe konzipiert. *Interdeutsch* und *TellMeMore* werden für erwachsene Sprachlernenden empfohlen. Nur ein einziges System hat somit Grundschüler als Zielgruppe. Moser schreibt im Jahre 1999, dass viele Lehrer der Meinung sind, dass computergestütztes Lernen auf Primarschule verfrüht sei (Moser, 1999:35). Diese Meinung hat sich seit dem Erscheinen des erwähnten Artikels wenig geändert hat, betrachtet man die Infrastruktur, die der Primarstufe zur Verfügung stehen. Im Kanton Aargau gibt es in der Regel zwei Computer in einem Schulzimmer. Computerräume, wie sie für die Oberstufe üblich sind und die von den Lehrpersonen für Unterrichtsstunden reserviert werden können, gibt es auf der Grundstufe nicht.

Auch wenn die Lerntheorien betrachtet werden, zeigt sich, dass traditionelle CALL-Anwendungen unterschiedliche Ansätze verfolgen. Entweder haben sie sich behavioristischen oder konstruktivistischen Ansätze verschrieben. Sowohl für behavioristische wie auch konstruktivistische Theorien steht ausführliches Feedback nicht an erster Stelle, weshalb diese Ansätze technisch einfacher zu implementieren sind.

Vergleicht man den Kontext der traditionellen CALL- mit demjenigen der ICALL-Anwendungen, fällt zuerst auf, dass nur gerade zwei ICALL-Systeme in der Praxis eingesetzt werden, die anderen vier sind Prototypen.

Von den sechs ICALL-Systemen sind der *e-Tutor* kommerziell und *ElditIV* frei verfügbar. Die anderen Anwendungen sind nicht über das Stadium des Prototypen hinausgekommen und werden heute nicht mehr weiterentwickelt. Eine Ausnahme ist der Prototyp *Robi*, welcher von Csilla Puskás im Rahmen ihrer Dissertation weiterentwickelt wird.

ICALL						
Kriterium	*e-Tutor*	*Bridge German Tutor*	*Eldit IV*	*Robi*	*Lice*	*Herr Kommissar*
Einsatz	kommerziell	Prototyp	frei verfügbar	Prototyp	Prototyp	Prototyp
L1/L2	L2	L2	L2	L2	L2	L2
Unterrichtsstufe	Erwachsene	Erwachsene	Erwachsene	keine bestimmte	Erwachsene	nicht eruierbar
Unterrichtstheorie	Kognitivismus	Kognititvismus	nicht eruierbar	keine bestimmte	Kognitivismus	Kognitivismus

Tabelle 5: ICALL-Anwendungen im Kontext

Betrachtet man die weiteren drei Parameter, die eine Anwendung im Kontext einordnen, zeigt sich, dass sich alle ICALL-Systeme sehr ähnlich sind. Keine einzige dieser Applikation wurde für den Erstsprachenunterricht konzipiert. Vier der sechs Anwendungen sind für erwachsene Lernende, seien das nun Studierende wie beim *e-Tutor* oder Soldaten wie beim *Bridge German Tutor,* konzipiert worden. ICALL-Systeme sind nicht auf verschiedene Zielgruppen zugeschnitten und es gibt keine Applikation, die auf Primarstufe oder Oberstufe eingesetzt werden könnte.

ICALL-Anwendungen sind entweder auf kognitive Theorien abgestimmt (*e-Tutor*, *Bridge German Tutor*, *Lice*, *Herr Kommissar*) oder es ist nicht klar, auf welchen Ansatz sie sich stützen. Wie in Abschnitt 4.1 erwähnt, versuchen ITS grundsätzlich den Ideen des Kognitivismus gerecht zu werden. Es erstaunt nicht, dass kognitive Ansätze bei den ICALL-Systemen sehr stark vertreten sind: die Forschung in den achtziger Jahren konzentrierte sich stark auf ITS. Die erwähnten Systeme wurden von dieser Forschung geprägt und drei davon wurden 1993 der Öffentlichkeit vorgestellt: *Lice* in der Doktorarbeit von Christopher Bowermann, der *Bridge German Tutor* und *Herr Kommissar* an

einem Workshop, der vom Army Research Institute finanziert wurde. Der *German Tutor,* der Vorgänger des *e-Tutors,* wurde zum ersten Mal im Jahre 2000 präsentiert. Allerdings forscht Trude Heift, die den *e-Tutor* heute kommerziell vertreibt und ihn weiterentwickelt, seit 1995 in diesem Bereich. Interessant ist, dass *Robi* ausdrücklich keiner Lerntheorie zugeordnet ist und nicht für eine bestimmte Lernstufe konzipiert wurde. Gemäss der Autorin der Anwendung war das primäre Ziel die linguistische Analyse von Lernereingaben und nicht, wie die Anwendung anschliessend eingesetzt wird.

Die vorangehende Diskussion zeigt zwei Punkte auf: erstens besteht eindeutig ein Mangel an intelligenten CALL-Lösungen für den L1-Unterricht auf der Primarstufe, weil sich die Entwickler von ICALL-Applikationen ausschliesslich auf erwachsene Lehrende konzentrieren. Zweitens, kommen gerade Systeme, die aus computerlinguistischer Sicht interessant sind, oft nicht über den Status des Prototypen hinaus (Gamper, 2003:339). Da sie als Prototypen konzipiert werden, fehlt ihnen der didaktische Kontext, der für den erfolgreichen Einsatz einer Anwendung im Unterricht wichtig ist. Das ist meiner Ansicht nach der Hauptgrund, weshalb ICALL-Anwendungen den Weg ins Klassenzimmer nicht finden. Im Bereich ICALL, gilt weiterhin was Rösler im 2004 geschrieben hat (Rösler, 2004:188):

> ICALL befindet sich schon seit längerer Zeit in einem Entwicklungsstadium, das noch weit davon entfernt ist, den Alltag des Fremdsprachenlernens mit digitaler Software zu bestimmen oder auch nur zu beeinflussen.

Speziell der *e-Tutor* beweist jedoch, dass intelligente und webbasierte CALL-Systeme technisch durchaus realisierbar sind und auch ausserhalb eines experimentellen Kontextes eingesetzt werden können.

5.3.2 Anwendungsqualität der untersuchten Applikationen

Um die Anwendungsqualität der untersuchten Anwendungen zu bestimmten, wurden sieben Aspekte untersucht: Übertragbarkeit, Wartbarkeit, Leistungsfähigkeit, Verlässlichkeit, Feedback, Übungstypen und Interaktion (vgl. Tabelle 2).

traditionelle CALL						
Kriterium	*Lernareal*	*LernNetz Deutsch*	*Schultraining*	*Interdeutsch*	*TellMeMore*	*German Steps*
Übertragbarkeit	webbasiert	webbasiert	webbasiert	webbasiert	webbasiert	webbasiert
Wartbarkeit	nicht möglich	nicht möglich	nicht möglich	nicht möglich	nicht möglich	nicht möglich
Leistungsfähigkeit	keine sichtbare Verzögerung	keine sichtbare Verzögerung	langes Aufstarten der Übungen	keine sichtbare Verzögerung	keine sichtbare Verzögerung	keine sichtbare Verzögerung
Verlässlichkeit	gut	gut	gut	gut	gut	gut
Feedback	teilweise Fehlerlokalisation, keine Analyse	teilweise Fehlerlokalisation, keine Analyse	keine Fehlerlokalisation & Analyse	teilweise Fehlerlokalisation, Alternativlösungen	keine Fehlerlokalisation & Analyse	teilweise Fehlerlokalisation, keine Analyse
Übungstypen	Richtig/Falsch, Multiple-Choice, Lückentext, Drag&Drop	Richtig/Falsch, Multiple-Choice, Lückentext, Drag&Drop	Richtig/Falsch, Multiple-Choice, Lückentext, Drag&Drop	Richtig/Falsch, Multiple-Choice, Lückentext, Drag&Drop	Multiple-Choice, Lückentext, Drag&Drop	Multiple-Choice, Lückentext
Interaktion	freie Auswahl der Übungen	freie Auswahl der Übungen	freie Auswahl der Übungen	freie Auswahl der Übungen	freie Auswahl der Übungen	freie Auswahl der Übungen

Tabelle 6: Anwendungsqualität von traditionellen CALL-Anwendungen

Tabelle 6 zeigt, dass sich traditionelle CALL-Applikationen in der Anwendungsqualität sehr ähnlich sind. Alle sechs traditionellen CALL-Systeme sind E-Learning-Anwendungen. Obwohl webbasierte Applikationen für die Lehrperson einfacher auf einer neuen Arbeitsstation einzurichten sind, gelten für alle obigen Anwendungen gewisse Einschränkungen: für *Lernareal*, *Interdeutsch*, *German Steps* muss ein Browser Javaskript, Flash und Popup-Fenster zulassen. *TellMeMore* kann ausschliesslich unter Internet Explorer angewendet werden.

Keine der traditionellen CALL-Systeme stellt den Lehrpersonen eine Möglichkeit zur Verfügung, neue Lerninhalte zu generieren. Das bedeutet, dass immer die gleichen Übungen abrufbereit sind. *Schulareal* z.B. stellt pro Wortart drei Übungen zur Verfügung, je eine einfache, mittlere und schwierige. Wenn alle drei Übungen zu einer bestimmten Wortart einmal gelöst sind, kennen die Schüler den Inhalt und die Motivati-

on ist gering, diese Übung nochmals zu lösen. Die Übungen solcher E-Learning-Applikation wirken dann monoton und langweilig. Das ist bedauerlich, weil die Anwendungen nützliche Übungstypen zur Verfügung stellen, mit denen Grammatikphänomene mehrmals geübt werden könnten.

Die Analyse der sechs traditionellen CALL-Anwendungen hat gezeigt, dass weder Leistungsfähigkeit noch Verlässlichkeit kritische Themen sind. Es dauert zwar länger bis die Übungen von *Schultraining* aufgestartet sind, anschliessend ist aber keine zeitliche Verzögerung zwischen Eingabe und Feedback bemerkbar.

Die sechs traditionellen CALL-Systeme nehmen keine Analyse einer falschen Eingabe vor. Ausser *Schultraining* und *TellMeMore* lokalisieren sie aber wenigstens fehlerhafte Eingaben. Das können sie, weil die Aufgabenstellungen sehr eingeschränkt sind. Bei restriktiven Anwendungen können die Schüler immer nur ein Wort in einen Lückentext eingeben oder ein bestimmtes Wort bei den Multiple-Choice-Aufgaben oder Auswählübungen anklicken. So ist von vornherein klar festgelegt, wo ein Fehler vorkommen kann. *Interdeutsch* schöpft zudem alle Möglichkeiten von *Hot Potatoes* aus, indem auch für falsche Lösungen Feedbacks hinterlegt wurden. Das ist allerdings für den Entwickler von *Interdeutsch* enorm zeitaufwändig, wenn er zusätzlichen Inhalt einpassen will.

Wenn man die Übungstypen betrachtet, welche die Anwendungen unterstützen, fällt auf, dass traditionelle CALL-Anwendungen vier der fünf Aufgabentypen einsetzen, die Schulmeister vorschlägt (siehe Abschnitt 3.2). Alle bieten Multiple-Choice-Übungen an; Richtig/Falsch-, Lückentext- und Drag- und Drop-Übungen werden von den meisten eingesetzt. Da *Hot Potatoes* genau diese Übungsarten unterstützt, ist es nicht erstaunlich, dass E-Learning-Anwendungen (*Lernareal*, *Lernnetz* und *Interdeutsch),* welche mit Hilfe dieser Autorensoftware geschrieben wurden, auch diese Übungsarten verwenden. Ob die anderen Anwendungen solche Übungen anbieten, weil sie bei den Lernenden po-

pulär sind oder weil sie technisch einfach zu realisieren sind, kann aufgrund dieser Analyse nicht bestimmt werden.

Bei allen traditionellen CALL-Systemen können die Schüler die Übungen aus einem Pool auswählen. Bis auf *German Steps* zeigen alle an, welche grammatischen Phänomene mit einer bestimmten Übung trainiert werden können. *Lernareal* und *Interdeutsch* bieten zusätzliche Auswahlmöglichkeiten über «einfach», «mittel» und «schwierig»; *LernNetz Deutsch* und *German Steps* haben die Übungen in Lerneinheiten organisiert, *Schultraining* hat sie nach Schulstufe geordnet. Bei *TellMeMore* ist unklar, ob die Übungen weiter strukturiert sind.

Betrachtet man hingegen die ICALL-Anwendungen unter den Aspekten der Anwendungsqualität fällt auf, dass speziell beim Feedback und den Übungstypen andere Wege gewählt wurden.

ICALL						
Kriterium	*e-Tutor*	*Bridge German Tutor*	*Eldit IV*	*Robi*	*Lice*	*Herr Kommissar*
Übertragbarkeit	webbasiert	nicht webbasiert	webbasiert	webbasiert	nicht webbasiert	nicht webbasiert
Wartbarkeit	nicht möglich	TUTOR	nicht möglich	nicht möglich	nicht möglich	nicht möglich
Leistungsfähigkeit	keine sichtbare Verzögerung	nicht eruierbar	keine sichtbare Verzögerung	Verzögerung <1s	nicht eruierbar	nicht eruierbar
Verlässlichkeit	gut	nicht eruierbar	gut	gut	bricht bei Falsch-eingabe ab	gut
Feedback	Fehlerlokalisation & -analyse	Fehlerlokalisation & -analyse	keine Fehlerlokalisation & -analyse	Fehlerlokalisation & -analyse	Fehleranalyse	Fehlerlokalisation & -analyse
Übungstypen	Lückentext, Diktat, Satzbau, Übersetzung	Multiple-Choice, Lückentext, Zuordnungsaufgabe, freie Eingabe	Lückentext, Drag&Drop, freie Eingabe	Lückentext, freie Eingabe	freie Eingabe	freie Eingabe
Interaktion	freie Auswahl der Übungen	vorgegebene Übungssequenz	freie Auswahl der Übungen	freie Auswahl der Übungen	nicht eruierbar	vorgegebene Übungssequenz

Tabelle 7: Anwendungsqualität von traditionellen CALL-Anwendungen

Bei den ICALL-Anwendungen finden sich drei Applikationen, die nicht webbasiert sind (*Bridge German Tutor*, *Lice* und *Herr Kommissar)*. Das bedeutet, dass die Lehrpersonen die entsprechende Software auf jeder Arbeitsstation neu installieren müssen. Demgegenüber muss bei webbasierten Lösungen jeweils nur die entsprechende Webseite

aufgerufen werden. Hier gilt allerdings zu beachten, dass nur *Eldit IV* und *Robi* unter jedem Browser einwandfrei funktionieren. Der *e-Tutor* hingegen kann nur unter Firefox oder Internet Explorer angewendet werden. Diese beiden Webbrowser werden am häufigsten verwendet[23]. Die Chance ist somit sehr gross, dass ein Anwender entweder Internet Explorer oder Mozilla Firefox als Browser zur Verfügung hat. Der *e-Tutor* kann daher von fast jedem Anwender benutzt werden.

Zusätzlicher Inhalt kann nur gerade für den *Bridge German Tutor* von den Lehrpersonen selbst erfasst werden. Bei den anderen Anwendungen müssen die Entwickler angefragt werden, ob sie neuen Inhalt hinzufügen. Die Entwicklung von ICALL-Applikationen ist speziell aufwändig und deshalb ist es sinnvoll, dass Lehrern Möglichkeiten zur Verfügung stehen, selbst neuen Inhalt zu generieren. Das Üben wird für die Schüler interessanter und die Anwendung kann längerfristig in den Unterricht eingebunden werden. Dies wiederum steigert die Akzeptanz der E-Learning-Anwendungen, sowohl bei den Schülern wie auch bei den Lehrpersonen.

Die Leistungsfähigkeit der ICALL-Anwendungen ist gut. Bei drei Anwendungen (*Bridge German Tutor*, *Lice* und *Herr Kommissar*) ist nicht eruierbar, wie lange es nach einer Eingabe dauert, bis ein Feedback ausgegeben wird, weil sie nicht getestet werden konnten. Die restlichen drei Applikationen weisen keine sichtbare zeitliche Verzögerung zwischen Input und Feedbackausgabe auf. Auch die Verlässlichkeit der Anwendungen ist gut, ausser bei den ICALL-Systemen *Lice* und *Robi.* Diese brechen bei gewissen fehlerhaften Eingaben ab, weil das Domänenmodell nicht vollständig ist. *Robi* kann z.B. nicht mit Fragesätzen oder Sätzen im Perfekt umgehen (Puskás; 2005:44). Die in *Lice* verwendeten Grammatiken können keine verschachtelten Strukturen analysieren (Bowerman; 1993:205). Wenn aber innerhalb der Möglichkeiten des Domänen-

23 Der Marktanteil des Internet Explorers beträgt 69,6%, gefolgt von Mozilla Firefox mit 24,6% (http://www.webhits.de/).

modells ein fehlerhafter Satz eingegeben wird, so nehmen die Systeme eine Analyse des Fehlers vor und unterstützen den Lernenden mit Rückmeldungen oder Grammatikhilfen. Alle ICALL-Anwendungen bis auf *Eldit IV* analysieren falsche Eingaben und alle – ausser *Lice* – lokalisieren auch die Fehler. Der *e-Tutor* hingegen weist – wie die traditionellen CALL-Anwendungen – sehr eingeschränkte Aufgabenstellungen auf und nimmt die Fehlerlokalisation auf diese Weise vor.

Auffallend ist, dass bei den ICALL-Anwendungen nur gerade der *Bridge German Tutor* und *Eldit IV* verschiedene Übungsarten anbieten (Multiple-Choice, Drag und Drop, Lückentext, freie Eingabe). Alle anderen ICALL-Systeme konzentrieren sich auf Lückentexte und Übungen mit freier Eingabe. Lückentextaufgaben können zwar ohne eine Analyse des Inputs verarbeitet werden, als Rückmeldung kann dann aber lediglich die korrekte Lösung eingeblendet werden (*Lernareal*, *Lernnetz)* oder der eingegebene Text wird mit Musterlösungen verglichen (*Interdeutsch*). Ein Input (egal ob innerhalb einer Lücke oder als freier Text) kann nur mit Hilfe von zusätzlichen morphosyntaktischen Informationen sinnvoll analysiert werden, und dazu braucht es NLP-Tools. Die Einbindung solcher Technologien ist der Fokus von computerlinguistisch motivierten Applikationen. Gleichzeit ist sie sehr aufwändig, was wohl erklärt, weshalb für ICALL-Applikationen weitere Übungstypen vernachlässigt werden.

Interessanterweise bieten nur die beiden ICALL-Systeme *e-Tutor* und *Eldit IV*, die heute auch ausserhalb der Forschung eingesetzt werden, Möglichkeiten, Übungen auszuwählen. Die Autoren von *e-Tutor* haben die Übungen nach Level und nach Lerneinheiten strukturiert, diejenigen von *Eldit IV* haben die Übungstexte nach Themen gruppiert. *Herr Kommissar* und der *Bridge German Tutor* schreiben eine Übungsabfolge vor. Bei *Robi* sind die Übungen nach grammatischen Phänomenen geordnet, welche

über Links angewählt werden können[24]. Die beiden ITS passen das Übungsprogramm den Schüler entsprechend deren Wissenstand an und stellen auf diese Weise eine Übungssequenz zusammen. Bei *Lice* ist unklar, ob und wie die Übungen strukturiert sind.

Der Vergleich von ICALL- und traditionellen CALL-Systemen unter dem Aspekt der Anwendungsqualität ist interessant, weil er einen klaren Gegensatz zwischen den beiden Softwarearten zeigt. CALL-Systeme sind zwar auf verschiedene und ganz spezifische Anwender ausgerichtet, unterstützten die Schüler aber nicht mit hilfreichen Rückmeldungen. ICALL-Anwendungen hingegen sind ausschliesslich auf erwachsene L2-Lerner ausgerichtet, fördern diese jedoch mit massgeschneidertem Feedback. Dieser Abschnitt hat auch gezeigt, dass es durchaus möglich ist, intelligente CALL-Anwendungen zu entwickeln, die nicht bezüglich Leistungsfähigkeit oder Verlässlichkeit eingeschränkt sind.

Es ist nicht erstaunlich ist, dass mir die Lehrpersonen keine ICALL-Anwendung für die Primarstufe empfohlen haben. Es gibt keine solche Applikation, die für den deutschen Grammatikunterricht auf dieser Stufe eingesetzt werden könnte. Die einzige traditionelle CALL-Anwendung, die von Grundstufenschülern benützt werden könnte (*Lernareal*), gibt keine intelligente Rückmeldungen aus und unterstützt die Schüler nicht im Lernprozess. Dies ist der Hauptgrund, weshalb *e-GramD* für die Primarstufe konzipiert wird und mit Hilfe von Annotationen intelligentes Feedback generiert wird. Die Diskussion in diesem Kapitel weist auf zwei Schlussfolgerungen hin, die für die Entwicklung von *e-GramD* unbedingt berücksichtigt werden müssen. Erstens muss den Lehrpersonen eine Möglichkeit geboten werden, zusätzlichen Lerninhalt einzugeben. Nur so kann die Anwendung in den Unterricht eingebunden werden und bleibt für die

24 In der Version von *Robi*, die in der Magisterarbeit von Puskà beschrieben wird, sind diese Links allerdings noch deaktiviert (Puskás; 2005:71).

Schüler interessant. Zweitens muss ein anwenderzentrierter Entwicklungsprozess gewählt werden, der Lehrpersonen einbindet. Das Problem von ICALL-Anwendungen liegt u.a. darin, dass die Möglichkeiten der künstlichen Intelligenz für CALL im universitären Rahmen getestet, dann aber – nach erfolgreicher Implementation eines Prototypen – nicht mehr weiterentwickelt werden. So sind die Systeme zwar von computerlinguistischem Interesse, weil sie aber häufig nicht ausserhalb der Universität eingesetzt werden, fehlt ihnen der Bezug zum Unterricht. Es trifft heute mehr denn je zu, "dass in der Entwicklung von ICALL in Zukunft Fremdsprachendidaktiker und Computerlinguisten enger zusammen arbeiten" (Rösler, 2004:190) müssen. Sonst bleibt der Einsatz von ICALL im Sprachunterricht tatsächlich Wunschdenken.

6 e-GramD

Die Diskussion in Abschnitt 4.3 hat ergeben, dass intelligentes Feedback für den Lernprozess nützlich ist. Der Vergleich von traditionellen CALL- und ICALL-Anwendungen in Kapitel 5 hat zudem gezeigt, dass eine webbasierte ICALL-Lösung für den Deutsch-Grammatikunterricht auf Primarstufe fehlt. *E-GramD* ist eine E-Learning-Anwendung, mit Hilfe derer Grundstufenschüler deutsche Grammatik trainieren können. Mit Hilfe eines Parsers (*Machinese Syntax* von Connexor) wird ein Korpus von Übungstexten analysiert, auf dessen Basis anschliessend Übungen und Feedback generiert werden. Abschnitt 6.1 beschäftigt sich mit der Auswahl dieses Parsers und der Notwendigkeit einer manuellen Qualitätskontrolle der von *Machinese Syntax* annotierten Texten. In Abschnitt 6.2 wird ein allgemeiner Überblick über die Systemarchitektur von *e-GramD* gegeben. In Abschnitt 6.3 wird das Modul zum Erstellen und Bearbeiten von Übungstexten beschrieben. Abschnitt 6.4 behandelt das Modul zum Lösen der Aufgaben.

6.1 NLP

Es wurde bereits mehrfach erwähnt, dass für die Generierung von intelligentem Feedback Sätze mit morphologischen Informationen versehen werden müssen. Das Ziel von *e-GramD* ist die Ausgabe von vernünftigen Rückmeldung, weshalb ein NLP-Parser benötigt wurde. Es wurden sechs verschiedene Systeme untersucht, die Texte analysieren und morphologische Information ausgeben. Drei Punkte waren ausschlaggebend bei der Evaluation: Erstens war es wichtig, über eine möglichst detaillierte Datenauswertung nach dem Parsevorgang zu verfü-

gen. Diese sollte nicht nur PoS-Informationen (Part-of-Speech) sondern auch Informationen zu Numerus, Kasus und Genus enthalten, weil genau diese Annotationen für sinnvolle Rückmeldungen benötigt werden. Zweitens ist für *e-GramD* eine Ausgabe in XML-Format ideal, da diese Datenstruktur einfach weiterverarbeitet werden kann (sowohl für die Übungsgeneration wie auch für die Rückmeldungen). Drittens musste ein Aufrufen des Parsers über die Kommandozeile möglich sein, damit der Parser in das Modul zum Erstellen und Korrigieren technisch integriert werden kann. Die folgende Tabelle gibt eine Übersicht über die verschiedenen Parser, die für diese Arbeit in Betracht gezogen wurden. Die mit einem * markierten Systeme wurden getestet, bei den anderen wurde die Evaluation aufgrund zu grosser technischer Hindernisse abgebrochen.

Name	*Kommandozeile*	*Ausgabeformat*	*Verfügbarkeit*
Morphy * *http://www.wolfganglezius.de*	nein	XML, Text	frei
Gertwol * *http://www2.lingsoft.fi/doc/gertwol/*	ja	Prolog, XML-ähnlich	kommerziell
SMOR *http://www.ims.uni-stuttgart.de/projekte/gramotron/SOFTWARE/SFST.html*	-	-	frei
CDG *http://nats-www.informatik.uni-hamburg.de/view/CDG/*	ja	-	frei
Durm *http://www.ipd.uka.de/~durm*	-	-	frei
Machinese Syntax * *http://www.conexor.fi*	ja	XML, Text	kommerziell

Tabelle 8: Übersicht über morphologische Parser

Morphy, *Gertwol* und *SMOR* sind rein morphologische Parser. Für *SMOR* wird ein kleines Lexikon mitgeliefert, welches erweitert werden müsste, um diesen Parser sinnvoll für *e-GramD* einsetzen zu können. *Morphy* und *Gertwol* basieren auf einem grossen Lexikon und könnten ohne zusätzlichen Aufwand eingesetzt werden. Das weitaus grössere Problem von exklusiv morphologischen Analysen ist, dass mehrere Lösungen pro Wort ausgegeben werden, weil keine syntaktische Auswertung vorgenommen wird. Das

bedeutet, dass morphologische Parser nicht entscheiden können, ob das Token «Blumen» Nominativ, Akkusativ, Dativ oder Genitiv Plural ist. Bei mehreren Parselösungen kann aber kein sinnvolles Feedback generiert werden. Eine intelligente Rückmeldung muss klar und eindeutig sein. Deshalb müssen die Parselösungen disambiguiert und mit möglichst detaillierten Annotationen versehen werden. Aus diesem Grund wurden *Morphy, Gertwol* und *SMOR* nicht weiter berücksichtigt und in der Folge Parser evaluiert, die nebst einer morphologischen Komponente auch eine syntaktische enthalten.

Das *Durm German Lemmatisation Project* und die *CDG* (Constraint Dependency Grammar Software) verfügen über beide Analyse-Komponenten und sind frei erhältlich. Obwohl die Auswertungen für die Rückmeldungen ausführlich genug sind, werden beide Systeme aus zwei Gründen nicht weiterverfolgt. Erstens ist unklar, in welcher Form die annotierten Texte ausgegeben werden. Zweitens ist die Installation beider Systeme sehr aufwändig und die mitgelieferten Lexika müssten ergänzt werden. Der zeitliche Aufwand dafür ist ausserhalb der Möglichkeiten des vorliegenden Projektes. *Machinese Syntax* der Firma Connexor erfüllt alle Kriterien, ist aber leider nur kommerziell verfügbar[25]. Der Parser kann dank zusätzlichen syntaktischen Informationen ambige Lösungen ausscheiden, wie das folgende Beispiel zum Wort «Blumen» zeigt:

1	Die	die	det>2	@PREMOD DET Def FEM SG NOM
2	Frau	frau	subj>3	@NH N FEM SG NOM
3	kauft	kaufen	main>0	@MAIN V IND PRES SG P3
4	Blumen	blume	obj>3	@NH N FEM PL ACC

«Blumen» wird richtig als Nomen im Akkusativ Plural erkannt und wird mit eindeutigen Annotationen versehen. *Machinese Syntax* ist ein Dependenzparser, der für jedes Token das Lemma, Part-of-Speech-Informationen, morphologische Merkmale und De-

25 Der Lizenzvertrag zwischen der Firma Connexor Oy und dem Institut für Computerlinguistik der Universität Zürich erlaubt ausschliesslich eine nicht kommerzielle, wissenschaftliche Verwendung der Software.

pendenzbeziehungen bestimmt. Es können sowohl individuelle Sätze wie auch ganze Texte analysiert werden. Die Ausgabe erfolgt entweder als reiner oder mit XML annotierter Text und der Parser kann über die Kommandozeile aufgerufen werden. Ausserdem ist er robust und kann mit Tippfehlern umgehen. In anderen Anwendungskontexten ist es wichtig, dass der Parsevorgang nicht abgebrochen wird, wenn eine falsche Struktur analysiert wird. Connexor hat *Machinese Syntax* deshalb so optimiert, dass auch falsche Sätze zerlegt werden. Der fälschlicherweise verwendete Artikel «die» im Satz «Ich habe die Versteck schon gefunden» wird als Akkusativ Plural bestimmt, obwohl «die» auch als weiblich Singular analysiert werden könnte. Solche Auswertungen sind aus pädagogischer Sicht nicht sinnvoll und können nicht für die Generierung von Rückmeldungen benutzt werden. Im Kontext von CALL-Anwendungen ist diese Art von Robustheit genauso problematisch wie die Präzision von 93,5% bei syntaktischen Analysen (Userguide, 2005:10). Mit anderen Worten: der Parser wertet in 6,5% der Fälle falsch aus. Die Rückmeldungen an die Schüler können somit auf keinen Fall ausschliesslich aufgrund der Ausgabe des Parsers ausgegeben werden. Diese beiden Problematiken gelten für jeden Parser, trotzdem basieren ICALL-Anwendungen (wie z.B. *Robi* oder *e-Tutor*) das Feedbacks auf den geparsten Daten. Anders als bei diesen Applikationen wird der Parser bei *e-GramD* genutzt, um ein Korpus von annotierten Übungstexten anzulegen, die in einem weiteren Schritt manuell kontrolliert werden. Aufgrund der korrekten Annotationen werden dann entsprechende Rückmeldung ausgegeben.

6.2 Systemarchitektur und Eigenschaften von *e-GramD*

E-GramD ist eine webbasierte Anwendung, die für das Training von Grammatikkompetenz im Deutschunterricht eingesetzt werden kann. Der Name *E-GramD* steht somit für webbasiertes (*e-*) Training von Grammatikkompetenzen (*Gram*) im Deutschunter-

richt (*D*). Mit Hilfe des Parsers *Machinese Syntax* von Connexor wird ein Korpus von Übungstexten analysiert, auf dessen Basis anschliessend Übungen und Feedback generiert werden. *E-GramD* besteht aus zwei Modulen: mit dem einen können Lehrpersonen Texte parsen und für die Übungen aufbereiten, mit dem anderen können Schüler anschliessend die Übungen lösen.

E-GramD ist so konzipiert, dass das Lösen von Übungen auch ohne Internetverbindung möglich ist. Dazu wurden die Übungen auf dem Client mit Javaskript geschrieben, weil mit dieser Skriptsprache Webseiten dynamisch manipuliert werden können. Trotzdem ist *e-GramD* als Client-Server-Architektur konzipiert, denn nur so können neue Übungen erstellt werden. Wird ein neuer Übungstext eingelesen, stehen zusätzliche Funktionalitäten zur Verfügung, die einen neuen Text parsen, für die Übungen aufbereiten und in einem Korpus speichern. Die nachfolgende Abbildung 7 visualisiert die Systemarchitektur von *e-GramD*.

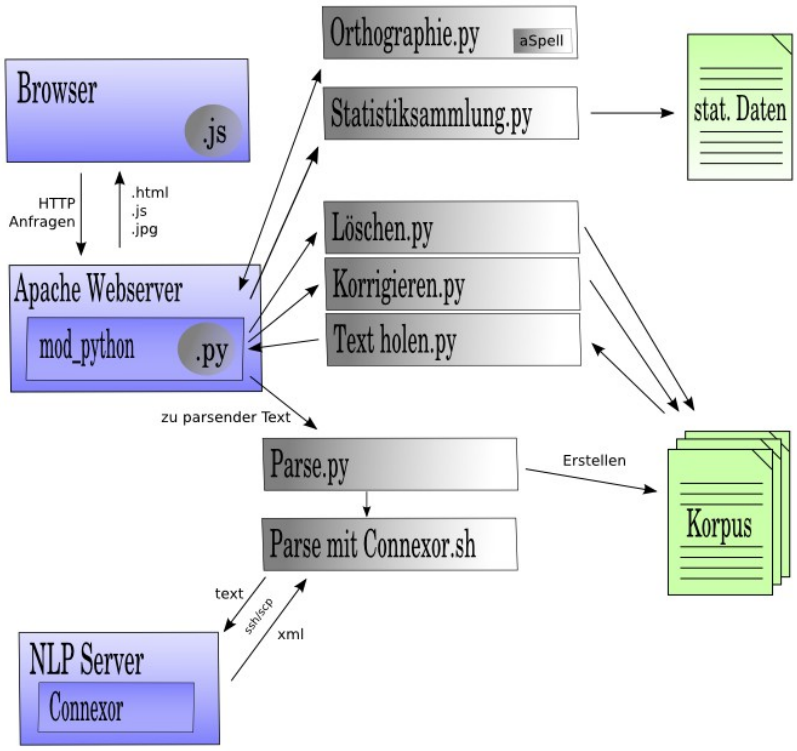


Abbildung 7: Systemarchitektur *e-GramD*

Es gibt fünf unterschiedliche Webseiten, die zum Erstellen oder Lösen der Übungen benötigt werden und die sich auf dem Client befinden. Von den fünf visuellen Oberflächen stellt eine Seite allgemeine Informationen zu *e-GramD* zur Verfügung, drei werden benötigt, um neuen Inhalt zu generieren und zu korrigieren, und eine visuelle Oberfläche dient der Aufgabenlösung. Die Webseiten sind in HTML und Javaskript programmiert. Mit Javaskript können Webseiten dynamisch manipuliert werden können, weshalb das Lösen der Übungen offline möglich ist. Für die Generierung von neuen Übungstexten werden allerdings über Pythonskripte verschiedene Aktionen ausgeführt. Damit Pythonskripte vom Webserver aufgerufen werden können, musste Mod_python in den Apache Webserver eingebungen werden. Somit wird beim Erstellen eines neuen Übungstextes ein Skript aufgerufen, das die NLP-Analyse auf dem Sprachserver[26] veranlasst. Die eingegebenen Sätze werden als reines Textfile an Connexor gesendet und nach dem Parsevorgang als .xml-File retourniert. Ein Pythonskript verarbeitet die XML-Annotationen so, dass sie für die Darstellung auf dem Client benutzt werden können (siehe Abschnitt 6.3). Über eine spezielle Webseite können die Annotationen überprüft werden. Ein Javaskript ruft auch hier Pythonskripte auf dem Webserver auf, welche die Rechtschreibeprüfung (*ASpell*) veranlassen und die Manipulation der Daten durch die Lehrperson zulassen. Schliesslich wird ein überprüfter Text mit Hilfe von Pythonskripten auf dem Mittelserver gespeichert.

Die Übungen selbst werden auf der Basis der annotierten Texte mit Javaskript direkt auf dem Client erstellt. Wie bereits erwähnt, können sie deshalb auch ohne Internetverbindung gelöst werden. *E-GramD* bietet vorläufig zwei Übungsarten an: Anklickübungen und Lückentexte[27]. Bei Lückentexten können nicht nur Grammatik-, sondern

26 Aus Lizenzgründen ist Machinese Syntax nur auf einem Server am Institut für Computerlinguistik installiert. Theoretisch könnte der Parser direkt auf dem Webserver installiert werden.

27 Die Funktionalitäten der beiden Übungstypen werden in Abschnitt 6.4 detailliert beschrieben.

auch Rechtschreibefehler vorkommen. Deshalb wird bei der Verarbeitung der Lückeneingaben eine Orthografieprüfung (*ASpell*) auf dem Datenserver vorgenommen. Wenn die Übungen offline gelöst werden, ist zu beachten, dass die Rechtschreibeprüfung zwar aufgerufen wird, die Daten aber nicht an *ASpell* übermittelt werden können und somit keine Hinweise zu falsch geschriebenen Wörtern gegeben werden. Ist ein Fehler nicht orthografischer Natur, wird ein Stringvergleich vorgenommen. Stimmt die Eingabe nicht mit der hinterlegten korrekten Lösung überein, wird ein spezifisches Feedback mit Javaskript generiert. Die Schüler können nach dem Korrekturvorgang mit der Maus über die fehlerhaften Eingaben fahren. In Sprechblasen oberhalb der falschen Eingaben wird ein detaillierter Hinweis zur korrekten Lösung angezeigt. Diese Rückmeldungen basieren auf den Annotationen, die bei jedem Wort hinterlegt sind. Welches Feedback aufgrund von welchen Annotationen erzeugt werden sollen, ist in einem Regelfile festgelegt. Nebst diesen spezifischen Rückmeldungen für jeden einzelnen Fehler wird auch eine allgemeine Auswertung der korrekten und falschen Antworten vorgenommen. Diese wird anschliessend in ein allgemeines Feedback verarbeitet. Für eine ausführliche Diskussion zum Thema Feedback von *e-GramD* wird auf Abschnitt 6.4 verwiesen. Schliesslich ist zu bemerken, dass bei jedem online Korrekturvorgang Daten über ein Pythonskript gesammelt und gespeichert werden. Diese Daten beinhalten Informationen zu den Eingaben der Schüler, die für eine spätere statistische Auswertung verwendet werden können.

6.3 Modul zum Erstellen und Bearbeiten von Lerninhalten

Das Modul zum Erstellen und Bearbeiten von Lerninhalten besteht aus drei unterschiedlichen Webseiten: erstens einer Übersicht über alle bestehenden Texte, zweitens einer Seite zur Eingabe von neuen Lerninhalten und drittens einer Seite, über welche die Annotationen der bereits geparsten Texten überprüft und gegebenenfalls korrigiert

werden können. Klickt die Lehrperson auf den Link «Texte bearbeiten», erhält sie eine Übersicht über alle bestehenden Texte.

Abbildung 8: Übersicht Übungstexte

Die Übersicht kann nur aufgerufen werden, wenn als Login-Name «Teacher» eingegeben wurde. Von dieser Übersichtseite aus können verschiedene Aktionen ausgewählt werden: Texte können gelöscht und die Korrektur- oder Eingabeseite aufgerufen werden. Die Löschfunktion wird dann benutzt, wenn z.B. ein bestimmter Text von allen Schülern gelöst wurde oder ein spezielles Thema durch ein anderes ersetzt werden soll. Beim Anwählen der Löschfunktion werden die Dateinamen der Texte mit dem Zusatz «.deleted» markiert und der Link im entsprechenden Dropdown-Menü auf der Webseite gelöscht. Ein Text wird nie gänzlich gelöscht, damit versehentliches Löschen zumindest durch einen Entwickler rückgängig gemacht werden kann. Für die Übersicht wird somit ein Filter angewendet, der alle Texte anzeigt, die keinen Zusatz «.deleted» aufweisen.

Bevor die Korrekturfunktion genauer erläutert wird, gilt die Aufmerksamkeit der Erfassung eines neuen Textes. Möchte eine Lehrperson neuen Lerninhalt hinzufügen, klickt sie auf den Button «Neuer Text».

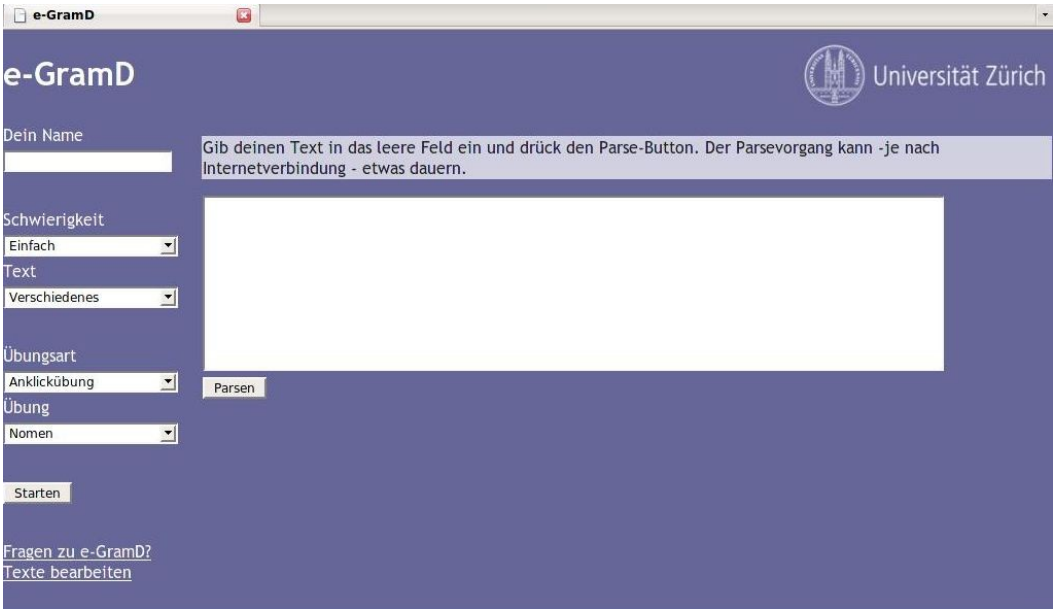


Abbildung 9: Einlesen eines neuen Textes

Die Lehrperson kann in das leere Textfeld einen beliebig langen Text eingeben und ihn dann parsen lassen. Die Auswertungszeit hängt von der Geschwindigkeit der Internetverbindung ab. Ein Lesetext der Primarstufe ist zwischen fünfzehn und dreissig Sätzen lang. Bei einer schnellen Verbindung dauern der Parsevorgang und das Laden der Übersichtsseite ungefähr 5.30 Sekunden. Dabei ist irrelevant, ob fünfzehn oder dreissig Sätze geparst werden. Die geparsten Texte werden von *Machinese Syntax* im XML-Format ausgegeben und in einem Korpus gespeichert. Damit die Annotationen für die Generierung der Übungen und der Rückmeldungen verwendet werden können, müssen die Daten aber mit Hilfe eines Pythonskripts nochmals aufbereitet werden. *Machinese Syntax* behandelt z.B. «am» als zwei Token und setzt die Präposition und den Artikel je in ein einzelnes Tag. Damit der Text in der Übung aber richtig angezeigt wird, darf zwischen diesen Tags kein Leerzeichen eingefügt werden. Ausserdem gibt *Machinese*

Syntax alle morphologischen Details in einem Tag <morpho> an, wie das folgende Beispiel für das Nomen «Fledermäuse» zeigt.

```
<token id="w2">
        <text>Fledermäuse</text>
        <lemma>fledermäuse</lemma>
        <depend head="w3">subj</depend>
                <tags>
                <syntax>@NH</syntax>
                <morpho>Heur N PL NOM</morpho>
                </tags>
</token>
```

Es muss einzeln auf PoS, Genus, Numerus, Kasus etc. zugegriffen werden, damit möglichst detaillierte Rückmeldungen generiert werden können. Trotzdem soll nur die für eine bestimmte Übung relvante Information im Feedback enthalten sein. Deshalb werden die morphologischen Informationen aufgesplittet und als einzelne Attribute des Tags behandelt. Nach der Aufbereitung sieht das Nomen «Fledermäuse» so aus:

```
<span type="word" id="w2" text="Fledermäuse" lemma="fledermäuse"
syntax="@NH" Heur="Heur" pos="N" num="PL"
cas="NOM">Fledermäuse</span>
```

Das XML-Element weist keine inhaltliche Struktur auf und beschreibt auch keine spezielle Form von Datenelementen. Es kann aber verwendet werden, um Abschnitte eines Textes auf eine bestimmte Art zu formatieren und besondere Attribute einzubinden. Ausserdem können Daten so einfacher mit Javaskript manipuliert werden.

Nachdem der Text geparst und für die Webseite aufbereitet ist, wird die Übersicht wieder geöffnet. Der neue Text erhält als Namen das heutige Datum und die Uhrzeit. Mit «Korrigieren» kann die Lehrperson den Namen des Textes ändern und die Annotationen überprüfen. Sie kann auch festlegen, welcher Schwierigkeitsstufe der Text zugeordnet werden soll. Wenn die Schwierigkeitsstufe nicht angepasst wird, ist der Text als

«einfach» eingestuft. In einem Textfeld wird der Text ohne Annotationen nochmals angezeigt.

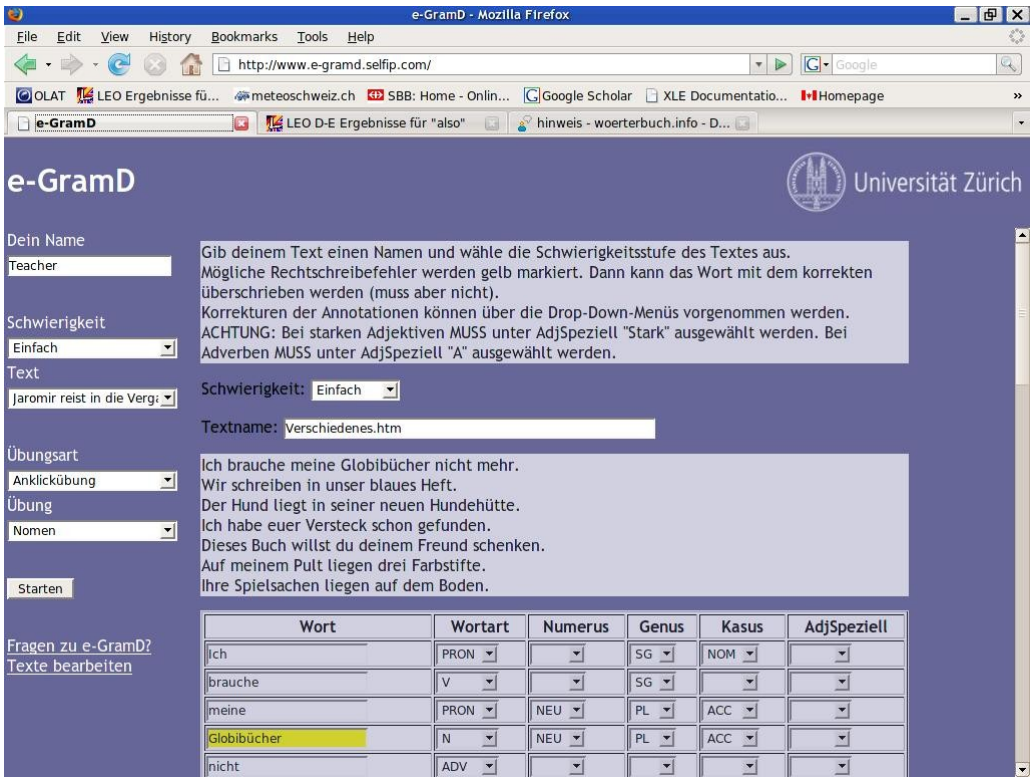


Abbildung 10: Korrektur eines geparsten Textes

In der Tabelle gleich unterhalb des Textfeldes können Orthografie, PoS[28] und morphologische Informationen angepasst werden. Wird ein Wort bei der Orthografieprüfung von *ASpell* nicht erkannt, wie im obigen Beispiel «Globibücher», wird es gelb markiert und die Lehrperson kann das Wort nochmals überprüfen. Diese Funktion wurde integriert, da während der Eingabe des Textes Tippfehler unterlaufen können. Part-of-Speech und morphologische Informationen können über Drop-Down-Menüs angepasst werden, wenn *Machinese Syntax* etwas falsch parst (siehe Abschnitt 6.1). «Meine» wird z.B. im Satz «Ich brauche meine Globibücher nicht mehr» als feminines Pronomen analysiert. Über das Drop-down-Menü kann die Lehrperson das Genus von «meine» auf

28 Damit sich auch Lehrpersonen ohne computerlinguistische Ausbildung zurechtfinden, wird statt Part-of-Speech "Wortart" verwendet.

sächlich setzen. In Abschnitt 4.2.2 wurde erwähnt, dass in der vierten Klasse alles als Adjektiv bestimmt wird, was gesteigert werden kann, ohne Rücksicht auf syntaktische Funktionen. *Machinese Syntax* unterscheidet jedoch zwischen Adjektiven und Adverben. Da diese Unterscheidung später doch noch eingeführt wird, werden die Annotationen von *Machines Syntax* belassen. Stattdessen wurde im -Element ein zusätzliches Attribut «AdjSpeziell» definiert. Dieses muss während der Korrekturphase bei allen Adverben angewählt werden. Das Attribut «AdjSpeziell» muss zudem auch bei allen starken Adjektiven eingefügt werden, weil *Machinese Syntax* nicht zwischen stark und schwach flektierten Adjektiven unterscheidet. Damit bei den Lückentexten aber keine Demonstrativpronomen vor starke Adjektive eingefügt werden können, ist es notwendig, dass diese beiden unterschieden werden (siehe auch Abschnitt 6.4.2).

6.4 Modul zum Lösen von Übungsaufgaben

Um Grammatikkompetenz für Deutsch zu trainieren, können die Schüler auf der Einstiegseite von *e-GramD* über die Navigationsleiste auf der linken Seite eine Übung auswählen. Auf Anraten der involvierten Lehrpersonen wurde das GUI einfach strukturiert und auf bewegte, blinkende oder tönende Elemente verzichtet, welche die Schüler beim Lösen der Übungen ablenken. Damit die Oberfläche dennoch für Viertklässler ansprechend ist, wurde die Darstellung eines Bücherwurms (Professor Würmli) für die Korrekturfunktion gewählt. Professor Würmli gibt den Schülern Anleitungen, wie sie eine Übung auswählen oder lösen können, aber auch Feedback zu einer gelösten Aufgabe. Ausserdem wurden alle Übungen mit einem Tier- oder Landschaftsfoto hinterlegt.

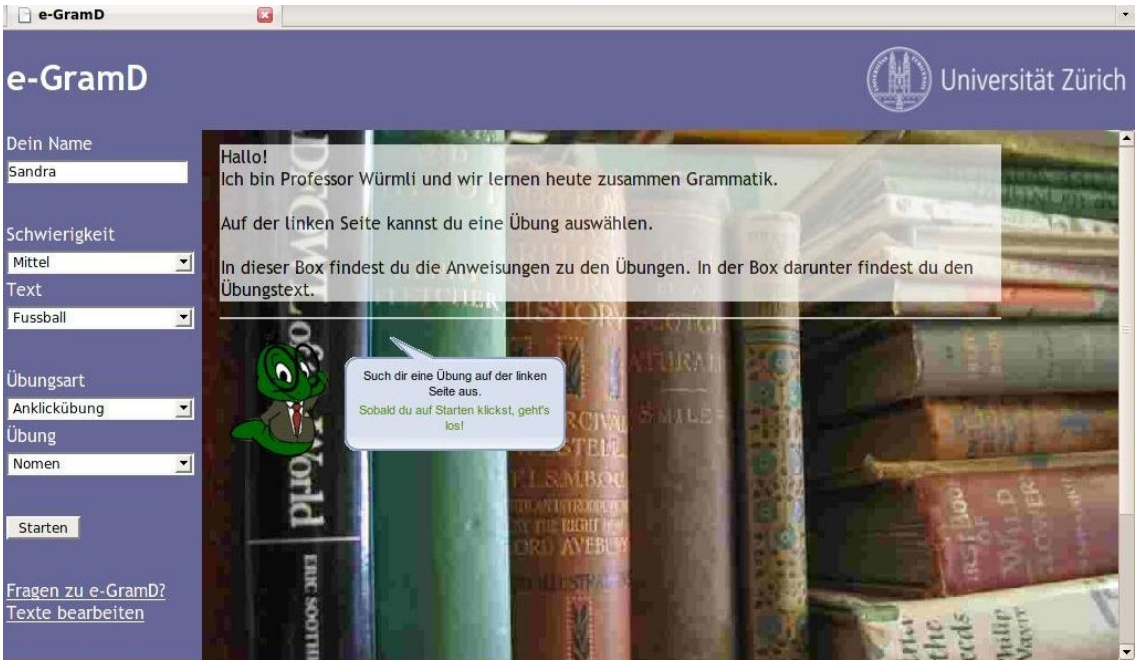


Abbildung 11: Einstiegsseite Übungen

Um eine Übung zu starten, müssen die Schüler ihren Namen eingeben und dann den Schwierigkeitsgrad wählen. Der Name muss eingetragen werden, damit die statistischen Daten, die bei jedem Korrekturvorgang an den Server geschickt werden, pro Schüler ausgewertet werden können. Die Schüler können über den Schwierigkeitsgrad (einfach, mittel und schwierig) auf Subkorpora von Übungstexten zugreifen. Zur Zeit besteht das Korpus aus fünf Texten. Ein Text basiert auf dem Lehrmittel *Grammatik,* die restlichen vier sind thematisch auf den Unterricht abgestimmt (siehe Diskussion Kapitel 4). Diese vier Texte wurden zusammen mit der Lehrperson ausgewählt und sind den Interessen der Schüler der Testklasse angepasst. Zwei Texte behandeln das Thema Fledermaus, welches im Fach Realien behandelt wurde; ein Text befasst sich mit Fussball und einer mit Schlittenhunden. Für jeden Text wurden alle Übungen generiert, so dass die Nomenerkennung zuerst an einfachen Texten geübt, später dann mit schwierigeren gefestigt werden kann. Die Schüler können über das Drop-Down-Menü «Übungsart» wählen, ob sie ein bestimmtes grammatisches Phänomen in einem

Text suchen (6.4.1) oder einen Lückentext (6.4.2) ergänzen wollen. Schliesslich können sie bestimmen, welche Wortart sie trainieren wollen, weil für jede Wortart alle Übungsarten implementiert worden sind.

6.4.1 Anklickübungen

Mit den Anklickübungen kann das Erkennen von Wortarten, Genus und Numerus trainiert werden. Übungen zum Kasus sind noch nicht vorhanden, da diese morphosyntaktische Kompetenz erst in der fünften Klasse erworben wird (siehe Abschnitt 4.2.1). Damit auch der Kasus trainiert werden könnte, müsste lediglich ein Entwickler die Datei «Anklickübung.js» im Modul *Excercises* ergänzen. Der Übungstext wird nach Auswahl eines Textes angezeigt, dabei können alle Wörter angeklickt werden. In der nachfolgenden Übung (Abbildung 12) müssen alle Nomen gefunden werden. Damit die Schüler die Nomen nicht aufgrund der Grossschreibung erkennen können, werden für diesen Übungstyp alle Wörter, bis auf diejenigen am Satzanfang, klein geschrieben.

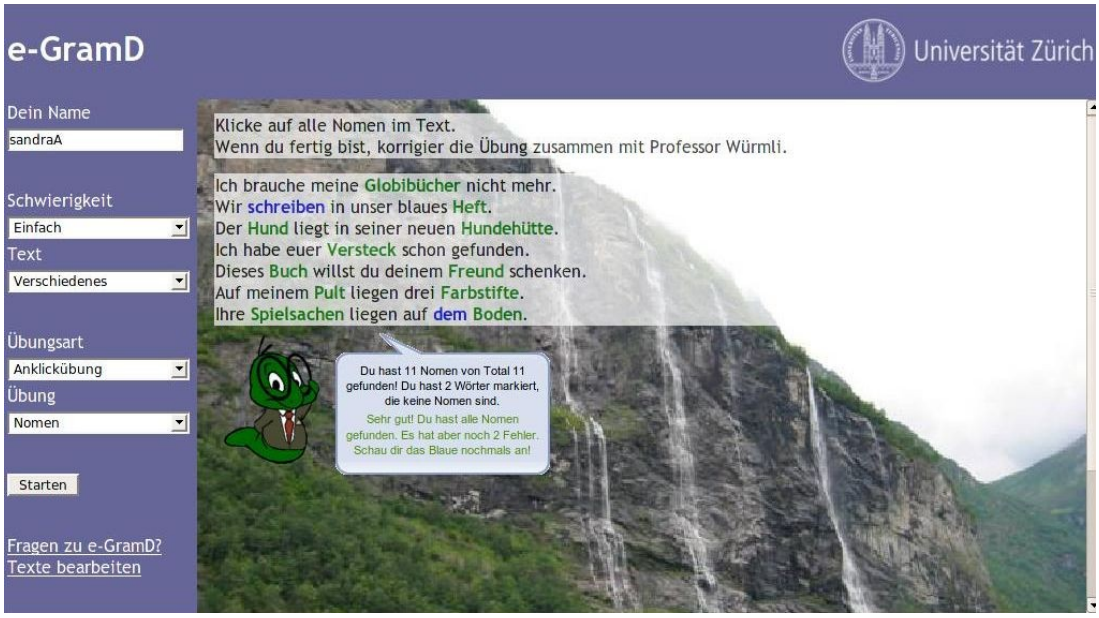


Abbildung 12: allgemeine Auswertung nach Korrektur

Sobald Professor Würmli um Korrektur gebeten wird, werden die korrekt markierten Wörter grün markiert, falsch angeklickte bleiben blau. Als erste Rückmeldung wird eine allgemeine Auswertung der gelösten Übung gegeben. Diese allgemeine Korrektur wird bei allen Übungen – unabhängig vom Aufgabentyp und dem grammatischen Phänomen – angezeigt. Diese Rückmeldung beschränkt sich auf die Angabe, wieviele Nomen bereits gefunden wurden, wieviele noch gesucht werden müssen und wieviele fälschlicherweise angeklickt wurden. Die Schüler können zu jedem beliebigen Zeitpunkt auf Professor Würmli anwählen, um herauszufinden, wieviele Nomen sie noch nicht gefunden haben. Wenn die Schüler über die blau markierten Wörter fahren, werden ihnen in einer Sprechblase detaillierte Informationen zum fälschlich ausgewählten Wort gegeben. Abbildung 13 zeigt die Rückmeldung, die präsentiert wird, wenn statt eines weiblichen Nomen ein sächliches markiert ist.

Abbildung 13: detaillierte Rückmeldung Anklickübung

Dazu werden drei Funktionen über Javaskript aufgerufen, welche die Informationen zu Genus und Part-of-Speech vom Token «Globibücher» natürlichsprachlich aufbereitet[29].

29 Die Funktionen «wordInfo» und «convert» sind nicht vollständig abgebildet, weil sie die Regeln für verschiedene morphologischen Informationen enthalten.

```
function ttNomen(word) {
        var wi = wordInfo(word);
        return {
                top:"Bist du sicher, dass das ein Nomen ist?",
                bottom:"Ich glaube, es ist ein " + wi.nicePos + "."
        };
}
function wordInfo(word){
        return {
        nicePos:convert(word.getAttribute('pos'),word.getAttribute('syntax')),
        };
}
function convert(pos, syntax){
        if (pos == 'N') return "Nomen";
        return "mir unbekanntes Wort";
}
```

Die Funktion «convert» enthält Regeln, über welche Attribute die Wortart definiert ist. Bei den Nomen ist diese Regel einfach, bei den Begleitern hingegen sind drei Regeln notwendig, damit dieser Kategorie nebst Bestimmungswörtern auch Pronomen und Numerale zugeordnet werden. Auch wenn weibliche Nomen gesucht werden, sind mehrere Regeln notwendig, um ein vollständiges Feedback im Tooltipp anzuzeigen.

6.4.2 Lückentexte

Die Lückentexte werden aufgrund von PoS-Informationen generiert. Dabei werden alle Token der Wortart, die geübt werden soll, ausgeblendet. Mit den Lückentextübungen können momentan Begleiter und Adjektive geübt werden. Diese beiden Arten wurden programmiert, weil die erste Wortart zu den geschlossenen, die zweite zu den offenen Klassen gehört und die Übungen unterschiedlich gestaltet wurden. Für die geschlossene Klasse der Begleiter wurde eine zusätzliche Datei mit Begleitern von *Machinese Syntax* geparst, manuell überprüft und abgespeichert. Diese Liste wird verwendet, um eine eingegebene Lösung aufgrund von Genus und Numerus abzugleichen. Deshalb kann beim Satz «Ich habe _____ Versteck schon gefunden» jeder positive Begleiter ein-

gegeben werden, der sächlich und in der Einzahl ist. Problematisch ist das Einsetzen des Begleiters «kein». Bei Sätzen wie «Ich brauche _____ Globibücher nicht mehr» darf dieser Begleiter nicht in die Lücke eingetragen werden, weil sonst doppelt negiert wird. Beim Satz «Wir schreiben in _____ blaues Heft» hingegen wird der Satz unsinnig, sobald «kein» eingesetzt wird. «Kein» ist daher nicht in der annotierten Datei enthalten. Für eine sinnvolle Überprüfung der Eingabe müsste nicht nur geprüft werden, ob der Satz bereits negativ ist, sondern auch, ob der eingesetzte Begleiter semantisch möglich ist. Eine weitere Herausforderung sind Sätze, bei denen ein Adjektiv auf den Artikel folgt, wie z.B. «Wir schreiben in _____ blaues Heft». In diesem Fall ist «blaues» stark flektiert und es können keine demonstrative Pronomen oder bestimmte Begleiter in die Lücke eingegeben werden. *Machinese Syntax* unterscheidet, wie bereits in Abschnitt 6.2 erwähnt, nicht zwischen starken und schwachen Adjektiven. Deshalb muss bei der Überprüfung der Annotationen bei starken Adjektiven ein zusätzliches Attribut "stark" eingefügt werden. Beim Lösen dieser Art von Übungen werden dann alle demonstrativen Pronomen und bestimmten Begleiter ausgefiltert.

Die Lückentexte für die offenen Klassen, wie es z.B. die Adjektive sind, werden anders aufgebaut. Für diese Klassen ist es nicht möglich, eine endliche Liste von möglichen Antworten zu generieren. Deshalb wird vor der Lücke das Lemma des Adjektives in Klammer angezeigt. Dadurch dass bei jedem Token nebst Angaben zu PoS auch die Lemma hinterlegt sind, kann einfach auf diese Information zugegriffen werden. Die Schüler müssen anschliessend die richtig flektierte Form in die Lücke eintragen.

Für die Lückentexte werden zwei Arten von Rückmeldungen ausgegeben, unabhängig ob es sich um Übungen für die geschlossenen oder die offenen Klassen handelt. Vorab wird eine Rechtschreibeprüfung durch *ASpell* vorgenommen. Wenn die Eingabe falsch geschrieben wurde, erhält der Lernende über die Sprechblase eine entsprechende Meldung.

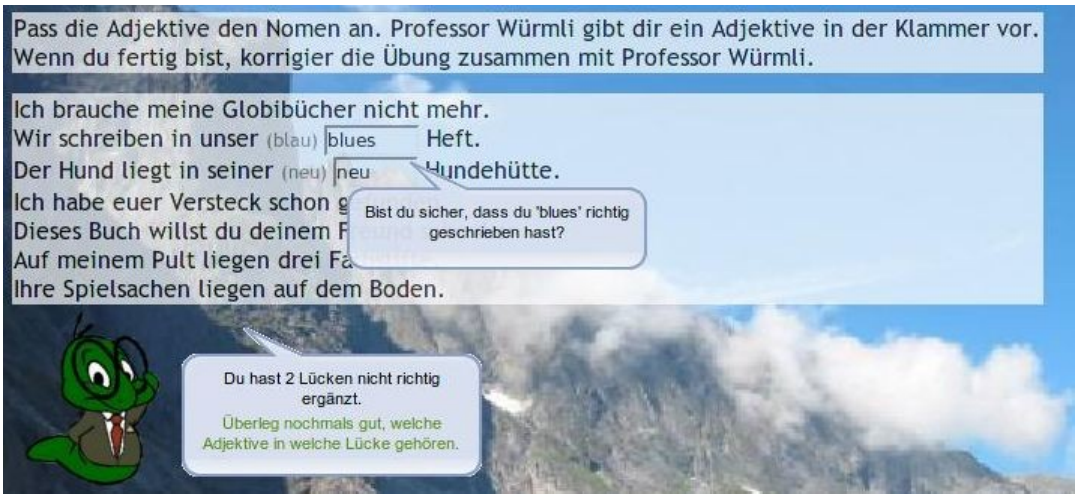


Abbildung 14: Feedback Rechtschreibefehler Lückentexte

Wenn das Wort richtig geschrieben ist, liegt ein Flexionsfehler vor. In diesem Fall gibt das Feedback Hinweise zum korrekten Genus und Numerus und hilft den Schülern auf diese Weise, die richtige Lösung zu finden (siehe Abbildung 15). Dieses Feedback wird auf der Basis der beim Wort korrekt hinterlegten Informationen generiert, im Gegensatz z.B. zum *e-Tutor*, bei welchem das Feedback aufgrund des Parsevorganges des falschen Satzes generiert wird (vergleiche Abschnitt 3.4).

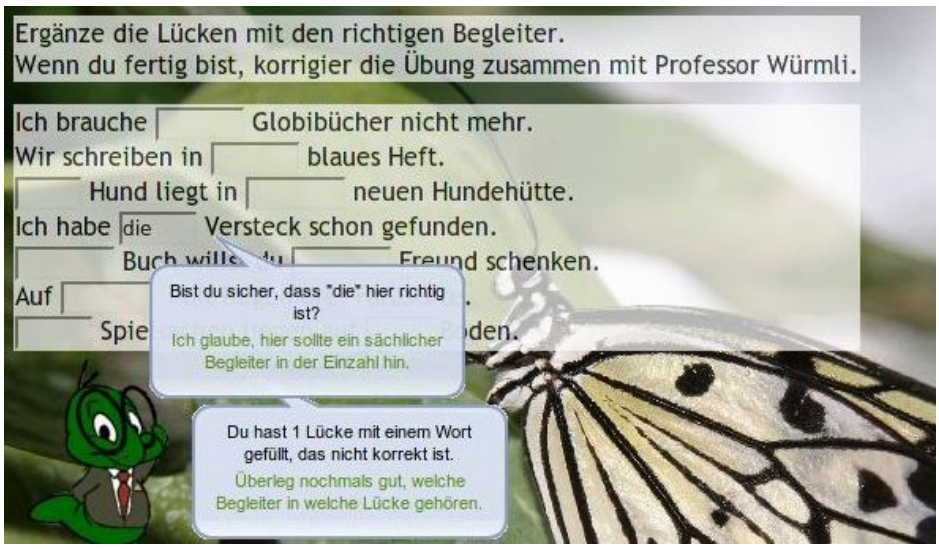


Abbildung 15: Feedback Deklinationsfehler Lückentexte

Auch die Rückmeldungen von *e-GramD* könnten aufgrund der Parseinformationen generiert werden. Wird der ganze Satz von *Machinese Syntax* geparst, so wird «die» als bestimmter Artikel im Plural analysiert. Der Lernende könnte «Versteck» aber auch den weiblichen Nomen zugeordnet haben. Daher ist es sinnvoller, Feedbacks aufgrund der korrekten Lösungen zu generieren, ansonsten basieren Rückmeldungen auf Entscheidungen, die ursprünglich gefällt wurden, um einen möglichst hohen Recall für den Parser zu erreichen.

7 Evaluation *e-GramD*

Die Evaluation der zwölf CALL-Anwendungen von Kapitel 5 hat gezeigt, dass traditionelle CALL-Anwendungen zwar auf eine Zielgruppe ausgerichtet sind, diese aber nicht mit intelligenten Rückmeldungen im Lernprozess unterstützen. ICALL-Applikationen hingegen helfen Lernenden mit Feedback, entsprechen aber nicht den Usability Standards. *E-GramD* wurde deshalb als webbasierte Applikation konzipiert, die intelligentes Feedback gibt. Im Rahmen der vorliegenden Arbeit sollte aber nicht nur eine E-Learning-Anwendung entwickelt, die Methoden der Computerlinguistik sinnvoll einsetzt, sondern auch evaluiert werden, wie eine solche Anwendung im Unterricht eingesetzt wird und wie sie von den Anwendern (Lehrpersonen und Schüler) akzeptiert wird. Aus diesem Grund wurde eine Evaluation von *e-GramD* durchgeführt. Dazu werden in Abschnitt 7.1 grundsätzliche Überlegungen zum Thema Evaluation von E-Learning-Anwendungen vorgestellt. Das restliche Kapitel ist drei unterschiedlichen Evaluationsphasen von *e-GramD* gewidmet. Erstens wird *e-GramD* in Abschnitt 7.2 mit dem Kriterienkatalog verglichen, der in Abschnitt 5.1 erarbeitet wurde. Zweitens werden in Abschnitt 7.3 die Resultate einer Evaluation von *E-GramD* durch 16 Grundstufenschülern vorgestellt. Drittens haben vier Lehrpersonen, die auf Primarstufe unterrichten, an einer qualitativen Befragung teilgenommen. Die Resultate dieser Untersuchung werden in Abschnitt 7.4 präsentiert. In Abschnitt 7.5 wird eine Synthese der drei unterschiedlichen Evaluationphasen durchge-

führt, um Stärken und Schwächen sowie das Entwicklungspotenzial von *e-GramD* einzuschätzen.

7.1 Evaluation von E-Learning-Anwendungen: Grundsätzliche Überlegungen

Im Bereich Evaluation von E-Learning-Software wird intensiv geforscht, wie die Fülle an theoretischer Literatur (vgl. Baumgartner, 1999; Fricke, 2003; Tergan, 2004) sowie die unzähligen tatsächlich durchgeführten Evaluationen (vgl. Nagata, 1996; Chi Yang, 1999; Gamper, 2002) zeigen. Den Autoren der theoretischen Forschung zufolge dienen Evaluationen der Qualitätsicherung. Baumgartner versteht unter dem Begriff Evaluation einen «Prozess der Beurteilung und Bewertung» (1999:1), aus dem am Ende ein klares Werturteil wie «Evaluand = gut/schlecht, wertvoll/wertlos» (1999:4) hervorgeht. Eine Evaluation hilft somit eine bestimmte Anwendung in die Fülle der CALL-Angebote einzuordnen. Dabei muss aber berücksichtigt werden, dass Beurteilungen unterschiedliche Funktionen haben können (Fricke, 2003:1). Sie können strategisch-politischen Entscheiden den Weg ebnen, wenn es z.B. grundsätzlich um die Einführung von E-Learning-Angeboten auf Primarstufe geht. Es ist auch möglich, dass sie als Entscheidungsgrundlage für die Entwicklung einer Applikation dienen. Schliesslich können sie eine Kontrollfunktion der Qualität einer Anwendung haben. Die vorliegende Evaluation von *e-GramD* beschäftigt sich mit dem letzten Punkt. Dabei ist es sinnvoll, möglichst unterschiedliche Perspektiven zu berücksichtigen, um die Qualität einer Anwendung umfassend zu bestimmen (Tergan, 2004:225) .

Je nach Interessen, die eine Evaluation verfolgt, wird sie unterschiedlich gestaltet. Das Team des schweizerischen Bildungservers SBS[30] ist im Begriff eine Lernsoftware-Datenbank aufzubauen (*Eva-Soft).* Im Rahmen dieses Projektes werden Lernsoftware-

30 http://www.educa.ch

Produkte für den Einsatz im Kindergarten und in der Volksschule bis zum 10./11. Schuljahr durch geschulte Lehrpersonen beurteilt.

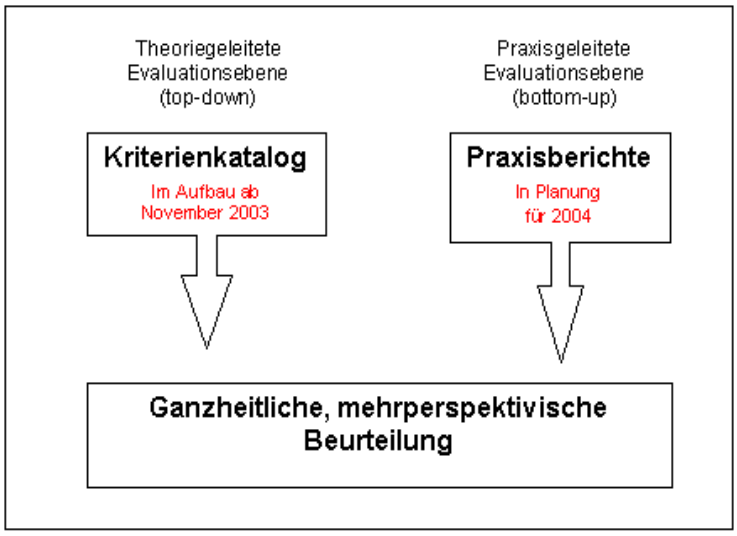


Abbildung 16: Konzept der Lernsoftware-Evaluation EvaSoft (http://www.educa.ch)

Dabei wird eine umfassende Evaluation angestrebt, indem mehrere Anwender den Unterichtseinsatz von Lernsoftware beurteilen und bewerten. Die Lehrpersonen analysieren die Software deshalb nicht nur anhand eines Kriterienkataloges, sondern informieren über den tatsächlichen Einsatz im Unterricht. Auch Tergan, Fischer und Schenkel basieren ihre webbasierte Evaluationsplattform *Eva*[31] auf denselben beiden Pfeilern, damit eine «umfassende Qualitätsevaluation» möglich ist (2004:225). Kriterienkataloge definieren technische und didaktische Qualitätsmerkmale. Das Produkt kann so technisch, didaktisch und medienpädagogisch in die Fülle von E-Learning-Anwendungen eingeordnet werden. Kriterienkataloge sind vor allem geeignet, um Produktemerkmale zu bestimmen, aber nicht um die Akzeptanz oder den Lernerfolg einer Anwendung zu messen (Tergan, 2004:224). Gemäss den Autoren des Evaluationsnetzes *Eva* wird der Lernerfolg durch eine «qualitativ hochwertige Lernsoftware und eine positive Akzeptanz» bestimmt. Die allgemeine Zufriedenheit der Anwender kann nur über eine Befra-

31 http://www.evaluationsnetz.de

gung derselben eruiert werden. Dabei müssen die Benutzerfreundlichkeit und die Nützlichkeit der Anwendung genauso wie der Gefallen am Programmdesign, am Inhalt und an den Funktionalitäten erfasst werden. Diese Aspekte sind keine zählbare Eigenschaften, die direkt quantitativ gemessen werden können. Über qualitative Befragungen können subjektive Erfahrungen von Anwendern hingegen erfasst werden. Auf diese Weise unterstützen sie die inhaltliche Einordung einer E-Learning-Anwendung. Es ist zu berücksichtigen, dass webbasierte CALL- oder ICALL-Applikationen zwei Benutzergruppen haben: Lehrpersonen und Schülern. Wenn eine Anwendung zwar Schüler anspricht, von Lehrpersonen aber nicht sinnvoll im Unterricht eingesetzt werden kann, fehlt die Akzeptanz auf Lehrerseite. In diesem Falle wird eine Applikation gar nicht erst in den Unterricht integriert. Wenn umgekehrt eine Software von Lehrpersonen als didaktisch sinnvoll beurteilt wird, Schüler diese aber als langweilig empfinden, wird die Zufriedenheit auf Schülerseite klein und somit der Lernerfolg gering sein. Lernerfolg hängt aber nicht alleine von der Akzeptanz der Anwendung durch die User ab. Wie Tergan, Fischer und Schenkel hervorheben, wird der erfolgreiche Einsatz einer E-Learning-Anwendung vor allem auch von Lernermerkmalen (Vorwissen, Motivation, Medienkompetenz und Interesse) und von strukturellen und personellen Rahmenbedingungen geprägt (Tergan, 2004:224). Die sechs Paramter zu den Rahmenbedingungen wurden in den Kriterienkatalog von Abschnitt 5.1 eingebaut.

Das Evaluationsnetz *Eva* gibt ferner noch drei weitere Ebenen einer Evaluation an: Transfer der Lerninhalte auf das Arbeitshandeln, Effizienz des Einsatzes und Kosten-Nutzen-Relation. Die Effizienz und Kosten-Nutzen-Relation sind im Rahmen dieser Evaluation nicht relevant, weil sie nicht kommerziell motiviert ist.

7.2 Usability von *e-GramD*

In diesem Abschnitt wird *e-GramD* den Evaluationskriterien, die in Abschnitt 5.1 erarbeitet wurden, gegenübergestellt und so die Usability der Anwendung kontrolliert. Die zwei Evaluationskriterien Bedienerfreundlichkeit und Funktionalitäten werden in diesem Abschnitt ausgeklammert, weil sie anderswo behandelt werden. Die Bedienerfreundlichkeit von *e-GramD* kann nur über die subjektive Erfahrung der Schüler bewertet werden. Deshalb wurden nach einer 30-minütigen Testphase die Eindrücke der Lernenden zu *e-GramD* gesammelt. Eine ausführliche Beschreibung der Antworten findet sich in Abschnitt 7.3. Die Funktionalitäten (Feedback, Übungstypen, Navigationsmöglichkeiten) von e-GramD wurden in Kapitel 6 ausführlich beschrieben.

Wie jede CALL-Anwendung ist auch *e-GramD* in einen Kontext eingebettet. Vorab wurden strukturelle Voraussetzungen geprüft, die den Einsatz von *e-GramD* bestimmen: Die Grundschulklassen der Gemeinde Nussbaumen verfügen über zwei Mac-Arbeitsstationen pro Schulzimmer. Die in dieses Projekt involvierte Lehrperson kann somit neue Medien lediglich als eine Möglichkeit der Wissensvermittlung nutzen (z.B. im Werkstattunterricht). Nicht nur in Nussbaumen (AG), sondern allgemein auf Primarstufe werden CALL-Anwendungen in den bestehenden Unterricht integriert. *E-GramD* wurde daher als Bestandteil eines «blended learning» Szenarios konzipiert. *E-GramD* ist frei verfügbar und kann im Erstsprachenunterricht verwendet werden. Vorläufig konzentriert sich die Anwendung auf Trainingsmöglichkeiten von deutschem Grammatikwissen. Die Zielgruppe sind Grundstufenschüler. Die Lerntheorie wurde durch das Lehrmittel *Grammatik* vorgegeben (siehe Kapitel 4).

Um die Übertragbarkeit sicherzustellen, wurden die technischen Voraussetzung an der Primarstufe in Nussbaumen AG vor der Entwicklungsphase geklärt. Dabei hat sich gezeigt, dass den Lehrpersonen kein Informatikdienst zur Seite steht. Das heisst, Lehr-

personen sind nicht nur Schulungsexperten und Autoren von neuem Inhalt, sondern auch Supporter. Die Grundstufenklassen verfügen jedoch über Internetzugang, weshalb *e-GramD* als E-Learning-Applikation konzipiert wurde. Wie bereits erwähnt, sind solche Anwendungen einfach zu installieren und die technische Hindernisse, um die Anwendung in den Unterricht zu integrieren, sind auf ein Minimum reduziert. Eine technische Voraussetzungen, um *e-GramD* im Unterricht einsetzen zu können, ist allerdings das Vorhandensein des Webbrowsers Mozilla Firefox. Dieser ist kostenfrei erhältlich, plattformunabhängig und ist auf allen Mac-Arbeitsstationen in Nussbaumen (AG) installiert.

Um eine gute Wartbarkeit zu garantieren, ist *E-GramD* in zweifacher Weise modular aufgebaut. Erstens wird der Lerninhalt über separate Webseiten erstellt respektive gelöscht. Der Lerninhalt kann auf diese Weise von den Lehrpersonen einfach gepflegt werden, so dass den Schülern regelmässig neuer Übungsinhalt zur Verfügung steht. Für eine ausführlichere Beschreibung, wie die Seite zum Erstellen mit derjenigen zum Lösen von Übungen verknüpft ist, wird auf Abschnitt 6.2 verwiesen. Die unterstützten Übungen sind einfach zu warten, genauso wie Anpassungen des Programmiercodes gut vorgenommen werden können. *E-GramD* setzt sich zusammen aus drei Programmier-Bausteinen und zwei Datensammlungen (mit * markiert):

Modul	*Verwendung*	*Modul*
html	visuelle Gestaltung Website	Erstellen, Lösen
Javaskript	Definition der Navigationsmöglichkeiten auf den Webseiten, der Übungen und des Feedbacks; Aufruf der Funktionen zum Erstellen und Korrigieren des Korpus	Erstellen, Lösen
Python	verschiedene Pythonskripts (für Aufruf des Parsers, Orthografieprüfung etc.)	Erstellen, Lösen
Images*	Sammlung von Hintergrundmotiven	Lösen
Texts*	Korpus der annotierten Texte	Erstellen, Lösen

Tabelle 9: Übersicht Module von *e-GramD*

Die visuelle Gestaltung der fünf Internetseiten wird im Modul *html* definiert. Alle Seiten weisen dasselbe Grundlayout und diesselbe Navigationsfläche auf der linken Seite auf. Über diese Leiste können entweder neue Texten eingelesen oder einzelne Übungen aus dem Aufgabenpool ausgewählt werden. Bei Aufruf einer Übung, wird im Hintergrund auf zufälliger Basis ein Tier- oder Landschaftsfoto angezeigt. Die Bildsammlung ist im Ordner *Images* gespeichert. Gemäss den Erfahrungen der involvierten Lehrperson spricht diese visuelle Gestaltung Primarschüler speziell an, ohne dass sie dadurch abgelenkt werden. Das Modul *Javaskript* enthält Javaskript-Funktionen, die für die Definition der Navigationsmöglichkeiten, des Layouts der Übungen und der Feedbacklösungen zuständig sind. Ausserdem enthält dieser Ordner alle Javaskripte, die für die Generierung und Korrektur von neuen Texten notwendig sind. Der Ordner *Python* enthält neben verschiedene Pythonskripts, welche das Speichern und Löschen der Korpustexte implementieren, sowie ein Unix Shellskript für den Parsevorgang. Zudem enthält das Modul Skripte, die zum Aufruf der Rechtschreibeprüfung eingesetzt werden. Das Korpus an annotierten Texten wird im Ordner *Texts* gespeichert.

Damit zwischen Input und Lösungsausgabe keine sichtbare Verzögerung eintritt, wurde *e-GramD* mit Javaskript programmiert. Diese Skriptsprache erlaubt die dynamische Anpassung einer Webseite, so dass beim Lösen der Übungen nicht auf den Server zugegriffen werden muss. Einzige Ausnahme bildet die Rechtschreibeprüfung, die über ein Pythonskript auf dem Mittelserver vorgenommen wird. Wie in Abschnitt 3.2 beschrieben, bezieht sich die Leistungsfähigkeit auch auf die Zeit, die ein Anwender benötigt, um die Anwendung richtig bedienen zu lernen. Deshalb wurde für alle visuellen Oberflächen die gleiche Navigationsliste benutzt. Die Schulungsdauer durch die Lehrperson wird damit auf ein Mimimum reduziert.

In Abschnitt 6.1 wurde bereits ausführlich erläutert, warum eine Qualitätskontrolle der Annotation von *Machinese Syntax* sinnvoll ist. Mit Hilfe der manuellen Korrektur

der geparsten Texte soll erreicht werden, dass die Rückmeldungen in jedem Fall korrekt sind. Somit ist die Verlässlichkeit von *e-GramD* sehr gut.

7.3 Resultate Evaluationsphase Schüler

E-GramD wurde von 16 Schülern während dreissig Minuten getestet und anschliessend bewertet. Die in Abschnitt 7.1 erwähnten Lernermerkmale werden im Folgenden genauer betrachtet. Alle Schüler der Testklasse hatten das gleiche Vorwissen; vor den Testlektionen wurden grammatische Phänomene der Nomen und Begleiter repetiert und diejenigen der Adjektiven eingeführt. Die Motivation und das Interesse der Schüler, *e-GramD* zu testen, war sehr hoch. Das rührt zum einen daher, weil computerbasierte Übungen von den Schülern lieber gelöst werden als traditionelle. Zum andern waren sie motiviert, weil sie explizit um ihre Meinung zur Applikation gebeten wurden. Die Medienkompetenz hingegen war sehr unterschiedlich; es gab Kinder, die zu Hause einen eigenen Computer besitzen, genauso wie solche, die keine Arbeitsstation privat zur Verfügung haben.

Für die Testphasen wurde nebst den beiden Mac-Computern ein Linux-Laptop benutzt. Damit alle Schüler an einer der drei Arbeitsstationen *e-GramD* testen konnten, wurde die Testphase während dem Halbklassenunterricht durchgeführt. Eine Gruppe bestand aus neun Schülern, die andere aus sieben. In beiden Gruppen waren sowohl Schülerinnen wie auch Schüler vertreten. Vorab wurden die Funktionalitäten von *e-GramD* anhand eines einfachen Übungstextes eingeführt. Anschliessend arbeiteten die Schüler[32] zu zweit oder zu dritt während dreissig Minuten an Übungen, die sie selbst auswählen durften. Danach wurden sie um ihre Meinung zu Inhalt und Funktionalitäten gefragt.

32 In diesem Kapitel wird Schüler auch in der Einzahl in der männlichen Form gebraucht. Damit soll verhindert werden, dass auf das Geschlecht der Lernenden rückgeschlossen werden kann.

Tergan, Fischer und Schenkel haben 45 Kriterien erarbeitet, mit denen die Akzeptanz der Benutzer gemessen wird (Tergan, 2004:228). Schüler der Primarstufe können einen solch ausführlichen Fragebogen nicht innerhalb eines sinnvollen Zeitrahmens ausfüllen. Der Fragebogen zur Teststunde enthielt – auf Anraten der involvierten Lehrperson – zehn Fragen, die sich auf den Inhalt und die Funktionalitäten beschränkte (siehe Anhang B). Die ausgewählten Fragen sind beeinflusst von der Usability Diskussion in Kapitel 3. Die erste Gruppe wurde von der involvierten Lehrperson als sprachlich stärker eingestuft als die zweite. Deshalb wurde bei der ersten Gruppe der Fragebogen verteilt und die Schüler konnten ihre Eindrücke zu *e-GramD* schriftlich festhalten. Bei der zweiten Gruppe hingegen wurde eine Gruppendiskussion zum gleichen Fragekatalog aufgezeichnet. Das Ziel war es, die subjektiven Eindrücke der Schüler zu sammeln. Drei Fragen erfassten die Übungstexte und -typen, welche die Schüler ausgewählt hatten. Je eine Frage betraf die Schwierigkeit des Übungsinhaltes und der Übungsart. Bei einer konnten die Schüler die Navigationsmöglichkeiten bewerten. Fünf Fragen berücksichtigten die allgemeinen Rückmeldungen und die spezifischen Hinweise zu fehlerhaften Eingaben. Die letzte Frage galt den zusätzlichen Wünsche und Anregungen der Schüler, was an *e-GramD* verändert oder ergänzt werden soll. Die Fragen waren alle offen gehalten, d.h. die Schüler konnten die Bewertungskriterien selbst bestimmen. Um einen Überblick über die Antworten zu geben, wurden diese für die nachfolgende Tabelle 10 in ein einheitliches Kriterienschema eingepasst, um die Evaluation veranschaulichen zu können. Das Ziel dieser Tabelle ist nicht eine Quantifizierung der Antworten sondern eine Visualisierung derselben.

No	**Inhalt**	**Kriterien**			
1	Textauswahl	Fledermäuse	Fussball		
		9/0	0/7		
2	Schwierig-keitsgrad Text	schwierig	mittelschwierig	einfach	
		1/0	7/5	1/2	
3	Übungswahl	Anklicküb ung	Lückentext		
		5/7	4/7		
4	Schwierig-keitsgrad Übung	schwierig	mittelschwierig	einfach	
		0/2	8/4	1/1	
5	Übungsaus-wahl	schwierig	mittelschwier	einfach	
		0/0	1/0	8/7	
6	Bewertung	sehr gut	gut	mittel	schlecht
		6/6	3/1	0/0	0/0
7	Verständnis Bewertung	sehr gut	gut	mittel	schlecht
		5/5	3/2	1/1	0/0
8	Kommentar	sehr gut	gut	mittel	schlecht
		3/7	6/0	0/0	0/0
9	Verständnis Kommentar	sehr gut	gut	mittel	schlecht
		3/5	6/2	0/0	0/0
10	Nützlichkeit Kommentar	nützlich	teilweise	nicht nützlich	
		8/6	0/0	1/1	

Tabelle 10[33]: Übersicht Evaluation Schüler

Die meisten Schüler wählten in den dreissig Minuten einen Text und dazu mehrere unterschiedliche Übungen aus. In der ersten Gruppe entschieden sich alle für einen Fledermaustext, in der zweiten alle für den Fussballtext. Der Fledermaustext wurde von der involvierten Lehrerin als schwierig eingestuft, die Schüler selbst bewerteten ihn unterschiedlich, und zwar von «einfach» bis «schwierig». Auch der Fussballtext wurde von der zweiten Gruppe unterschiedlich bewertet: von einfach bis mittelschwierig. Bei den Übungstypen wurde in der ersten Gruppe fünf Mal die Anklicküb ung und vier Mal der Lückentext ausgewählt. Die Schüler der zweiten Gruppe hatten offensichtlich mehr

33 Die roten Zahlen stehen für die Rückmeldungen der ersten Gruppe, die grünen für diejenigen der zweiten.

Zeit, konnten doch alle mehrere Anklickübungen sowie Lückentexte lösen. Die Schwierigkeitsgrade der Übungen wurden wiederum sehr unterschiedlich empfunden (von «einfach» bis «schwierig»). Die Navigation innerhalb des Übungspool selbst empfanden die Schüler, bis auf eine Person, als einfach. Die allgemeine Bewertung, die von Professor Würmli ausgegeben wurde, wurde als gut bis sehr gut eingestuft. Die Schüler verstanden diese allgemeinen Rückmeldungen grundsätzlich gut bis sehr gut. Die Kommentare von Professor Würmli zu den einzelnen Fehlern bewerteten zwei Schüler als sehr gut, die restlichen als gut. Die Kommentare wurden in der Regel gut bis sehr gut verstanden und haben den meisten Schülern geholfen, die richtige Lösung zu finden. In der zweiten Halbklasse wurden die Feedbacks als besonders nützlich bewertet, weil sie eine Hilfestellung bieten, um «die korrekte Lösung zu finden, diese aber nicht einfach anzeigen» (S11). Ausserdem unterstützen sie darin, herauszufinden, was genau an einer Lösung «falsch ist» (S10). Allerdings halfen die speziellen Hinweise zu den Fehlern zwei Schülern überhaupt nicht, ihre falschen Eingaben zu korrigieren. Die Schüler brachten unterschiedliche Anregungen ein, was an *e-GramD* verändert werden könnte. So sollten mit dieser Plattform andere Sprachen (z.B. Englisch, Französisch etc.) gelernt werden können. Ausserdem sollte *e-GramD* mehr Übungstexte und zusätzliche Übungstypen enthalten. Ein Schüler wünschte sich sogar, dass für Mathematik eine ähnliche webbasierte Anwendung programmiert würde. Eine weitere Anregung war, dass auf Wunsch eine Bewertung in Noten ausgegeben wird. Auch sollte Professor Würmli beim Lösen der Übung zusätzlich helfen, indem er z.B. den ersten Buchstaben eines gesuchten Begleiters in die Lücke einträgt. Ein anderer Schüler möchte gerne, dass die Übungstexte auf die Hintergrundbilder abgestimmt sind.

7.4 Resulate Evaluationsphase Lehrpersonen

Vier Lehrpersonen der Primarstufe haben *e-GramD* getestet und das Lösungsmodul schriftlich bewertet. Eine Lehrperson war bereits im Vorfeld in die Konzeption und das Design von *e-GramD* miteinbezogen worden und hatte daher ein grösseres Vorwissen. Den Lehrpersonen wurde der Link von *e-GramD* elektronisch zugeschickt. Sie erhielten keine Einführung zur Programmbedienung. Als Basis der Befragung diente ein Fragebogen, allerdings war es den Lehrpersonen freigestellt, wie stark sie auf die Fragen eingingen. Diese offene Art der Befragung erlaubte den Teilnehmenden, dass sie ihre persönliche Perspektive einzubringen. Der Fragekatalog kann in Anhang C eingesehen werden. Er enthält Fragen zur GUI-Gestaltung, zur Navigation innerhalb von *e-GramD*, zum Inhalt der Korpustexte, den Übungstypen und den Rückmeldungen. Ausserdem wurde nach den möglichen Einsatzszenarien gefragt. Ziel dieser Befragung sind wiederum keine quantitativ untermauerten Aussagen. Vielmehr dient sie dazu, die Akzeptanz von *e-GramD* von Seiten der Lehrpersonen abzuschätzen.

Das Design der Plattform wurde als ansprechend für die Primarstufe bewertet. Der Aufbau wurde mit dem Prädikat «gut strukturiert» versehen. Es wurde bemängelt, dass Schriftgrösse, Schriftbild und Zeilenabstand für die dritte Klasse zu klein sind. Auf dieser Stufe haben die Schüler noch Mühe mit engen und kleinen Schriftzeichen. Für die vierte Klasse sind diese Kriterien aber sinnvoll erfüllt. Die Hintergrundbilder sind attraktiv für die Primarstufe, sollen aber auf das Thema des Textes angepasst sein. Sonst könnten sie für die Schüler verwirrlich sein. Die Schulungsdauer wurde von allen Lehrpersonen als kurz eingeschätzt; sie glauben, dass die Schüler die Übungen, nach einer kurzen mündlichen Einführung, problemlos alleine lösen können. Der Begriff «Schwierigkeit» soll in der Navigationsliste vermieden werden, weil er auf die

schwächeren Schüler entmutigend wirken kann. Ausserdem soll klar definiert werden, auf welcher Basis ein Text in einen Schwierigkeitsgrad eingestuft wird.

Die Übungsinhalte wurden als spannende und kindergerechte Sachthemen bezeichnet. Die Texte zu den Schlittenhunden und Fledermäusen wurden als anspruchsvoll für die vierte Klasse eingestuft, der Fussballtext als mittelschwer. Der einfache Übungstext ist gemäss einer Lehrperson zu einfach für diese Schulstufe. Allerdings würde sie die einfachen Texte trotzdem behalten, damit auch Kinder, die noch nicht lange in der Schweiz sind, die Möglichkeit haben, mit *e-GramD* Grammatikphänomene des Deutschen zu lernen. Die Übungstypen und bestehenden Übungen dazu wurden als benutzerfreundlich beschrieben. Es wurde aber gewünscht, die Anklickübungen um eine Möglichkeit zu ergänzen, ab einer bestimmten Anzahl korrekt gefundener Lösungen alle Nomen, Begleiter oder Adjektive anzeigen zu lassen. Zu langes Suchen langweile die Schüler, so dass sie keine Lust mehr hätten, mit *e-GramD* zu arbeiten.

Die allgemeinen Bemerkungen von Professor Würmli wurden als gut eingeschätzt, weil die richtigen Antworten gezählt wurden und der Erfolg bewertet werde. Durch die zusätzlichen Angaben, wieviele falsche Eingaben gemacht wurden, unterstützt *e-GramD* die Schüler in der selbständigen Korrektur. Die detaillierten Hinweise von Professor Würmli über die falschen Eingaben wurden als sehr gut und differenziert eingestuft. Einer Lehrperson hat dies besonders gut gefallen. Sie arbeitet bereits mit *Lernareal* und kennt diese Art ausführlicher Rückmeldungen nicht.

Alle Lehrpersonen würden *e-GramD* ergänzend im Unterricht einsetzen, wenn sie die entsprechenden Grammatikthemen besprächen. Einerseits würden sie im Unterricht Zeit einplanen, um mit *e-GramD* zu arbeiten. Andererseits würden sie Übungen der Plattform als Hausaufgabe aufgeben. Das heutige Angebot an Übungen ist allerdings zu beschränkt, als dass *e-GramD* regelmässig verwendet werden könnte. Das Korpus

müsste um zusätzliche Texte ergänzt werden. Darüber hinaus müssten neue Übungen programmiert werden: So sollte der Übungspool z.B. um zusätzliche Lückentexte, mit denen die Nomenbildung geübt werden könnte, oder Anklickübungen zu Verben ergänzt werden.

Zusätzlich möchten die Lehrpersonen, dass die Schüler über ihren Namen bei einer Übung wieder da einsteigen könnten, wo sie beim letzten Mal aufgehört haben. Dies ist wichtig, weil E-Learning in den traditionellen Unterricht eingebettet ist und Übungen aus Zeitgründen nicht immer fertig gelöst werden können. Über den Namen sollte den Schülern am Ende einer Übungsabfolge auch ein Bericht angezeigt werden, der ausgibt, was erfolgreich gelöst wurde und was speziell verbessert werden müsste.

7.5 Synthese der drei Evaluationsteile

E-GramD wurde evaluiert, um die Qualität der Anwendung zu kontrollien. Deshalb bestand die Evaluation von *e-GramD* aus drei Teilen: dem Vergleich der Software mit einem Kriterienkatalog, den Kommentaren von Schülern nach einer halbstündigen Testphase und dem kritischen Input durch Lehrpersonen.

Der Vergleich von *e-GramD* mit dem Kriterienkatalog hat gezeigt, dass die Anwendung den technischen Anforderungen einer guten E-Learning-Anwendung entspricht. Speziell hervorzuheben sind hier die zusätzliche Autorenseite, um weitere Texte zu generieren. Zudem verbessert die manuelle Qualitätskontrolle der Annotation die Verlässlichkeit des Feedbacks. Dafür ist die Übertragbarkeit von *e-GramD* nicht optimal. Die Applikation wurde ursprünglich für Mozilla Firefox konzipiert, weil dieser plattformunabhängig ist. Allerdings hat sich nach der Testphase herausgestellt, dass die Schüler zu Hause häufig den Internet Explorer benützen. Das wiederum bedeutet, dass nicht alle Schüler Hausaufgaben mit *e-GramD* lösen können. Die Applikation müsste daher für

zwei Webbrowser optimiert werden: wie der *e-Tutor* sollte sie unter Internet Explorer und Mozilla Firefox laufen.

Die Testphase durch die Schüler hat demonstriert, dass die Funktionalitäten von *e-GramD* einfach zu erlernen sind. Nach einer kurzen Einführung konnten alle Schüler selbständig am Computer Übungen lösen. Die Schüler haben die Anwendung gut akzeptiert; sie fanden das Layout der Seite ansprechend und die Übungstexte und -arten spannend. Alle Schüler wollten von zu Hause aus weiter mit *e-GramD* üben, was leider aus Lizenzgründen nicht möglich ist. Die E-Learning-Anwendung hat aber nicht nur die Schüler angesprochen, sondern auch die Lehrpersonen. Sie glauben, dass die Applikation für die intendierte Zielgruppe sinnvoll im Unterricht eingesetzt werden kann. Obwohl die Lehrpersonen keinerlei Schulung zu *e-GramD* erhalten hatten, fanden sich alle damit zurecht. Das zeigt, dass die Funktionalitäten von *e-GramD* einfach zu erlernen sind und somit die Schulungszeit auf ein Minimum beschränkt ist. Die technische Hürde ist für Lehrpersonen minimal, um *e-GramD* in den Unterricht zu integrieren.

Die Feedbacks, die über eine rein numerische Angabe hinausgingen, haben nur gerade bei einer Lehrpersonen Erstaunen ausgelöst. Dies hängt wohl von zwei Punkten ab: erstens setzen die Lehrpersonen CALL-Anwendungen spärlich auf der Grundstufe ein. Ausserdem kannte keine Lehrperson die Möglichkeiten von ICALL, so dass intelligente Hinweise für sie eine neue Funktionalität waren. Für sie standen vor allem didaktische Fragen im Vordergrund. So möchten z.B. alle Lehrpersonen, dass die Schüler eine abschliessende Bewertung der gelösten Übungen erhalten. Diese Übersicht soll entweder ausgedruckt oder an die Lehrperson gemailt werden. Eine solche Auswertung hilft nicht nur den Schülern, sondern zeigt den Lehrpersonen, welches Thema im Unterricht nochmals aufgeriffen werden muss.

Diese Evaluation basiert auf qualitativen Auswertungen der erhobenen Daten. Dazu wurden die Einschätzungen von vier Lehrpersonen analysiert sowie die Rückmeldungen der Schüler nach der Testphase. Aufgrund dieser Auswertung können die Vor- und Nachteile von *e-GramD* bestimmt werden. Auch kann abgeschätzt werden, inwiefern diese Art von ICALL-Systemen auf Primarstufe sinnvoll ist. Die Schlussfolgerungen dürfen aber nicht quantitativ betrachtet werden, weil sechzehn Schüler und vier Lehrpersonen eine zu kleine Benutzergruppen sind. Gerade bei den Lehrpersonen sind sehr unterschiedliche didaktische Überzeugungen in die Kommentare eingeflossen. Deshalb müsste *e-GramD* unbedingt von einer grösseren Anwendergruppe von Lehrpersonen evaluiert werden. Es wäre auch sinnvoll, dass mehr Schüler die Anwendung testen. Um eine valide Aussage machen zu können, ob die Sprachkompetenz durch den Einsatz von *e-GramD* tatsächlich verbessert wird, müsste die Applikation zudem über längere Zeit mit den Leistungen im Unterricht verglichen werden. Dadurch, dass bei jeder Eingabe der Schüler statistische Daten gespeichert werden, bietet *e-GramD* die Möglichkeit, eine solche Evaluation durchzuführen. Weiter wäre es auch sinnvoll, den Lernerfolg von *e-GramD* mit demjenigen einer traditionellen CALL-Anwendung zu vergleichen. Zum jetzigen Zeitpunkt können jedoch keine gültigen Schlussfolgerungen zur Frage, ob die Sprachkompetenz durch den Einsatz von *e-GramD* gesteigert wird, gezogen werden.

8 Zusammenfassung und Ausblick

Ziel dieser Lizenziatsarbeit war die Entwicklung einer E-Learning-Anwendung für den Deutschunterricht auf Primarstufe (*e-GramD*) unter Verwendung von computerlinguistischen Methoden, welche die Schüler bei der Aufgabenlösung durch spezifisches und korrektes Feedback unterstützt. Zudem sollte mit *e-GramD* zusätzlicher Lerninhalt generiert werden. Dadurch wird der Lerninhalt zwar nicht gerade in eine praxisrelevante Situation eingebettet, wie es von den Konstruktivisten gefordert wird, doch immerhin können so die speziellen Interessen der Schüler berücksichtigt werden. Ausserdem erleichtert diese Funktionalität von *e-GramD* den Lehrpersonen die Generierung von neuen Übungen. Um diesen zwei Anforderungen gerecht zu werden, ist der Einsatz eines NLP-Parsers notwendig. Statt selbst einen Parser zu implementieren, der keine vollständige Grammatik enthält, wurde auf einen in der Praxis erfolgreich erprobten Parser zurückgegriffen: *Machinese Syntax* von Connexor ist ein Parser mit hohem Recall und guter Precision. Dessen ungeachtet, annotiert er teilweise falsch. Das Konzept von *e-GramD*, mit Hilfe eines Parsers ein Korpus annotierter Übungstexte aufzubauen, die anschliessend manuell kontrolliert werden, hat sich gut geeignet, um mit dieser Problematik umgehen zu können.

Desweiteren sollte *e-GramD* so konzipiert sein, dass die Anwendung in den traditionellen Unterricht der Grundstufe integriert werden kann. Deshalb wurde das Design der Anwendung stark auf die Zielgruppe abgestimmt und bei didaktische Fragen eine Lehrperson mit in den Entwicklungsprozess einbezogen. Über eine

Testphase in einer Klasse und die Einschätzungen von vier Lehrpersonen wurde evaluiert, ob eine solche Anwendung überhaupt in den bestehenden Grammatikunterricht integriert würde. Obwohl die Evaluanden die Anwendung grundsätzlich positiv bewertet haben, gibt es in diesem Bereich noch einige Punkte zu verbessern, damit ein längerfristiger Einsatz von *e-GramD* garantiert ist. Erstens muss *e-GramD* für den Internet Explorer optimiert werden. Zweitens müssen zusätzliche Aufgaben zu den beiden Übungsarten generiert werden. Drittens muss *e-GramD* in ein Gesamtkonzept eingebettet werden, mit dem auch andere Fähigkeiten trainiert werden können (Hörverständnis, Lesetraining, etc.). Diese drei Punkte wären der Inhalt zukünftiger Forschungsprojekte.

CALL-Anwendungen haben sich seit den sechziger Jahren technisch stark weiterentwickelt; waren sich die Applikationen ursprünglich alle sehr ähnlich, gibt es heute eine grosse Vielfalt an Angeboten. Nebst traditionelle CALL-Anwendungen finden sich auch ICALL-Applikationen, die von computerlinguistischem Wissen Gebrauch machen. Auch sind Systeme heute sehr häufig multimedial und webbasiert. Auffallend ist, dass sich die meisten ICALL-Anwendungen für ausführliche Rückmeldungslösungen entschieden haben, während sich traditionelle Anwendungen mit «richtig» respektive «falsch» als Antworten auf falsche Eingaben begnügen. Ganz selten werden auch alternative Rückmeldungen hinterlegt, weil die Definition von zusätzlichen Lösungsalternativen sehr zeitaufwändig ist. Ausserdem ist es unmöglich, dass ein Entwickler alle möglichen Fehler durch zusätzliche alternative Feedbacks abdecken kann. ICALL-Anwendungen lösen die Problematik der intelligenten Rückmeldungen elegant, indem sie mit Hilfe von Parsern eine Fehlerlokalisation und -analyse vornehmen. Heute liegt bei der Entwicklung von Parser der Fokus auf dem Recall. Parser werden also möglichst robust konzipiert, damit sie bei Falscheingaben nicht abbrechen. Gerade diese Art von Robustheit erweist sich für ICALL-Anwendungen aber als hinderlich. Für E-Learning-Anwen-

dungen steht nicht der Recall sondern die Precision im Vordergrund. Solange aber das Feedback direkt im Anschluss an den Parsevorgang generiert wird, können Rückmeldungen fehlerhaft sein. Wenn Schüler falsche Informationen zu ihren Eingaben erhalten ist das erstens didaktisch nicht sinnvoll. Zweitens ist die Akzeptanz solcher Anwendungen von Seiten der Lehrpersonen niedriger. Genau aus diesem Grund werden ICALL-Anwendungen nicht in den Unterricht integriert.

Obwohl eine technische Diversifikation des Angebotes stattgefunden hat, sind die Zielgruppen von traditionellen CALL- und ICALL-Anwendungen immer noch hauptsächlich Oberstufenschüler oder erwachsene Lernende. Trotzdem legen die heutigen Applikationen den didaktischen Fokus sehr unterschiedlich. Das ist durchaus sinnvoll, weil je nach Sprachkompetenz, die vermittelt werden soll, unterschiedliche Anforderungen an die Applikation bestehen. Traditionelle CALL-Anwendungen basieren hauptsächlich auf dem behavioristischen Ansatz, Entwickler von ICALL-Applikationen lehnen diesen in der Regel ab und orientieren sich an kognitivistische Paradigmen. Wenn Sprachkompetenz für den Alltag vermittelt werden soll, wie es das Ziel im L2-Unterricht ist, ist der behavioristische Ansatz tatsächlich nicht passend. Behavioristische Applikationen haben aber durchaus ihre Berechtigung, wenn es z.B. – wie im Deutschunterricht auf Primarstufe – in erster Linie darum geht, grammatische Kategorien bewusst zu machen. Dazu eignen sich die behavioristischen Pattern-Drill-Übungen gut. Solche Aufgaben basieren aber oft auf allgemeinen Texten aus einem Übungsbuch, welche nicht auf die Interessen der Schüler eingehen. Im traditionellen Unterricht ergänzen Lehrpersonen solche Textbuchaufgaben teilweise manuell um zusätzliche Übungen. Es ist für Lehrpersonen jedoch sehr zeitaufwändig, wenn sie die Grammatikübungen auf Papier auf andere Unterrichtsthemen anpassen müssen. CALL kann gerade auch in diesem Bereich unterstützen, indem eine Anwendung den Lehrpersonen eine Möglichkeit gibt, selbst neue Übungen zu generieren. Dazu eignen sich Autorensoftwa-

res nicht, weil die Lernphase in der Regel zu lange ist. Ein Sonderfall ist *Hot Potatoes:* die Software ist schnell und einfach zu erlernen. Trotzdem dauert die Übungserstellung viel zu lange, weil alle alternativen Lösungsmöglichkeiten einzeln definiert werden müssen, wenn die Rückmeldungen detailliert sein sollen.

Abschliessend kann festgehalten werden, dass der Einsatz von NLP traditionelle CALL-Anwendung bereichert. Parseinformationen können für die Generierung von intelligentem Feedback genutz werden. Allerdings müssen die Grenzen von NLP klar erkannt werden und – wie bei *e-GramD* – Möglichkeiten für eine manuelle Qualitätskontrolle geschaffen werden. Trotzdem darf sich die Forschung von ICALL nicht nur auf die technische Machbarkeit konzentrieren. Die vorliegende Arbeit hat gezeigt, dass ICALL-Anwendungen Entwickler zwar vor gewisse Herausforderungen stellen, diese technisch aber durchaus zu lösen sind. ICALL-Konzepte müssen heute vielmehr in einen realen Unterrichtskontext eingebettet werden, damit die entwickelten Applikationen später auch genutzt werden. Hier bestätigt die vorliegende Arbeit mehrfach, was Rösler (2004) in seinem Werk *E-Learning Fremdsprachen – eine kritische Einführung* verschiedenfach betont: ICALL braucht den Input von Fremdsprachendidaktik genausowie die technische Expertise von Computerlinguisten.

Literaturverzeichnis

[AARG. LEHRPLAN 2005] Kanton Aargau (2005). *Fachlehrplan Deutsch Volksschule*. http://www.ag.ch/lehrplan/de/pub/lehrplan_volksschule/faecher/test.php [Stand: 1.1.2007]

[BAUMGARTNER 1999] Baumgartner, Peter (1999). «10 Todsünden in der Evaluation interaktiver Lehr- und Lernmedien». *Studieren 2000 – Alte Inhalte in neuen Medien?* Lehmann, Klaus, Hrsg. Münster: Waxmann, 199–220.

[BAUMGARTNER 2002] Baumgartner, Peter, Hartmut Häfele und Kornelia Maier-Häfele (2002). *E-Learning Praxishandbuch; Auswahl von Lernplattformen: Marktübersicht – Funktionen – Fachbegriffe*. Innsbruck: Studienverlag.

[BEAZLEY 2001] Beazley, David M. (2001). *Python Essential Reference*. 2. Ausgabe. Indiana: New Riders.

[BEVAN 1999] Bevan, Nigel (1999). «Quality in use: Meeting user needs for quality». *Journal of Systems and Software* 49(01): 89–96.

[BEVAN 2001] Bevan, Nigel. «International standards for HCI and usability». *Int. J. Human-Computer Studies* (55): 533–552.

[BÖRNER 2001] Börner, Wolfgang et al., Hrsg (2001). *Grammatik lehren und lernen: didaktisch-methodische und unterrichtspraktische Aspekte.* Bochum: AKS-Verlag.

[BOWERMAN 1993] Bowerman, Christopher (1993). *Intelligent Computer-Aided Language Learning LICE: A System to Support Undergraduates Writing in German*. Ph.D. thesis, UMIST, Manchester.

[BURGER 2005] Burger, Harald (2005). *Mediensprache. Eine Einführung in Sprache und Kommunikationsformend der Massenmedien*. 3. Auflage. Berlin: de Gruyter, 32-63.

[CHI YANG 1999] Chi Yang, Jie und Kanji Akahori (1999). «An Evaluation of Japanese CALL Systems on the WWW Comparing a Freely Input Approach with Multiple Selection». *Computer Assisted Language Learning* 12 (1): 59–79.

[CHAMBERS 1983] Chambers, Jack A. und Jerry W. Sprecher (1983). *Computer-Assisted Instruction: Its Use in the Classroom*. New Jersey: Prentice-Hall.

[CONNEXOR 2005] Connexor Oy (2005). *Userguide Machinese Linguistic Analysers*. Connexor.

[DONATH 1998] Donath, Reinhard (1998). *Deutsch als Fremdsprache, Projekte im Internet*. Stuttgart: Klett.

[FELIX 2004] Felix, Uschi (2004). «Orchestrated Vision». *Language Magazine* (10/04): 24–29.

[FLÜCKIGER 2004] Flückiger, Max und Peter Gallmann (2004). *Richtiges Deutsch: Vollständige Grammatik und Rechtschreibelehre unter Berücksichtigung der aktuellen Rechtschreibereform*. 26. Aufl. Zürich: Verlag Neue Zürcher Zeitung.

[FRICKE 2003] Fricke, Reiner (2003). «Methoden der Evaluation von E-Learning-Szenarien im Hochschulbereich». Vortrag im Rahmen des Expertenworkshops Evaluation. Tübingen: Institut für Wissensmedien. http://www.tu-braunschweig.de/Medien-DB/empaed/evaluation-von-e-learning.pdf

[GAMPER 2002] Gamper, Johann und Judith Knapp (2002). «A Review of Intelligent CALL Systems». *Computer Assisted Language Learning* 15 (4/02): 329–342.

[GREENE 2004] Greene, Cara, Katrina Keogh, Thomas Koller, Joachim Wagner, Monica Ward und Josef van Genabith (2004). «Using NLP Technology in CALL». *NLP and Speech Technologies in Advanced Language Learning Systems - Proc.* of InSTIL/ICALL2004 Symposium on Computer Assisted Language Learning. Delomonte, Philippe Delcloque und Sara Tonelli, Hrsg. Venedig, Italien, 55–58.

[HEIFT 2001a] Heift, Trude (2001a). «Intelligent Language Tutoring Systems for Grammar Practice». *Zeitschrift für Interkulturellen Fremdsprachenunterricht* [Online] 6(2): 1–15.
http://www.spz.tu-darmstadt.de/projekt_ejournal/jg_06_2/beitrag/heift2.htm
[Stand: 14.06.2007].

[HEIFT 2001b] Heift, Trude und Devlan Nicholson (2001b). «Web Delivery of Adaptive and Interactive Language Tutoring». *International Journal of Artificial Intelligence in Education* 12: 1–14.

[HEIFT 2006] Heift, Trude (2006). «Context-sensitive *Help* in CALL». *Computer Assisted Language Learning* 19 (2&3): 243–259.

[HOCHSTRASSER 2006] Hochstrasser, Monika und Ursula Trauffer (2006). *Grammatik 4. Klasse.* Luzern: Lehrmittelverlag Trauffer & Hochstrasser.

[KERRES 2001] Kerres, Michael (2001). *Multimediale und telemediale Lernumgebungen: Konzeption und Entwicklung*. 2. Aufl. München: Oldenbourg.

[KREYER 1995] Kreyer, Steve und Eleanor Criswell (1995). «Instructor as Author in an Adaptive, Multimedia, Foreign Language Tutor». *Intelligent Language Tutors. Theory Shaping Technology.* Holland, Melissa V., Jonathan D. Kaplan und Michelle R. Sams, Hrsg. New Jersey: Lawrence Erlbaum, 45–54.

[LUTZ 2004] Lutz, Mark und David Ascher (2004). *Learning Python*. 2. Ausgabe. Sebastopol: O'Reilly Media.

[LEVY 2006] Levy, Yair (2006). *Assessing the Value of E-Learning Systems*. Hershey: IGI.

[MITSCHIAN 1999] Mitschian, Haymo (1999). *Neue Medien – neue Lernwerkzeuge: Fremdsprachenlernen mit Computern; Erfahrungen und Möglichkeiten für Deutsch als Fremdsprache*. Bielefeld: Bertelsmann.

[MOSER 1999] Moser, Heinz (1999). «Den Einstieg hab ich gefunden, die Meisterschaft noch nicht. Zwischenbericht des Projektes 'Multimedia in Primarschulen'» http://www.baselland.ch/docs/ekd/schulen/eval/computer_primar.pdf [Stand: 26.01.2007].

[NAGATA 1993] Nagata, Noriko (1993). «Intelligent Computer Feedback for Second Language Instruction». *The Modern Language Journal* 77 (3): 330–339.

[NAGATA 1996] Nagata, Noriko (1996). «Computer vs. Workbook Instruction in Second Language Acquisition». *CALICO Journal* 14 (1): 53–75.

[NANDORF 2003] Nandorf, Katja (2003). «Interaktivität in multimedialer Englisch-Lernsoftware für erwachsene Selbstlerner». *Fremdsprachenlernen mit digitalen Medien*. Legutke, Michael und Dietmar Rösler, Hrsg. Tübingen: Gunter Narr, 33–63.

[NEWHAM 1998] Newham, Cameron und Bill Rosenblatt. *Learning the Bash Shell*. 2. Ausgabe. Sebastopol: O'Reilly Media.

[ONG 2000] Ong, Jim und Sowmya Ramachandran (2000). «Intelligent Tutoring Systems: The What and the How.», http://www.learningcircuits.org/2000/feb2000/ong.htm [Stand: 27.06.2007]

[POWERS 2007] Powers, Shelley. *Learning JavaScript*. Sebastopol: O'Reilly Media.

[PUSKÁS 2005] Puskás, Csilla (2005). «Interaktives E-Learning im Bereich Deutsch als Fremdsprache». http://www.uni-giessen.de/~gm1160/ma/abstract.html, Magisterarbeit Justus-Liebig-Universität, Giessen.

[RAY 2003] Ray, Erik T. (2003). *Learning XML*. 2. Ausgabe. Sebastopol: O'Reilly Media.

[RÖSLER 2004] Rösler, Dietmar (2004). *E-Learning Fremdsprachen – eine kritische Einführung*. Tübingen: Stauffenburg.

[SAMS 1995] Sams, Michelle R. (1995). «Advanced Technologies for Language Learning: The BRIDGE Project Within the ARI Language Tutor Program». *Intelligent Language Tutors. Theory Shaping Technology*. Holland, Melissa V., Jonathan D. Kaplan und Michelle R. Sams, Hrsg. New Jersey: Lawrence Erlbaum, 7–21.

[SCHULMEISTER 2002] Schulmeister, Rolf (2002). *Grundlagen hypermedialer Lernsysteme: Theorie – Didaktik – Design*. 3. Aufl. München: Oldenbourg.

[SEGERMANN 2001] Segermann, Krista (2001). «Wie kann Grammatik gelernt werden?». *Grammatik lehren und lernen: didaktisch-methodische und unterrichtspraktische Aspekte.* Börner, Wolfgang et al., Hrsg. Bochum: AKS-Verlag, 193–207.

[TERGAN 2004] Tergan, Sigmar-Olaf, Arno Fischer und Peter Schenkel (2004). «Qualitätsevaluation von E-Learning mit dem *Eva*luationsnetz». *Evaluation von E-Learning. Zielrichtungen, methodologische Aspekte, Zukunftsperspektiven*. Meister, Dorothee M., Sigmar-Olaf Tergan, Peter Zentel, Hrsg. Münster: Waxmann, 223–233.

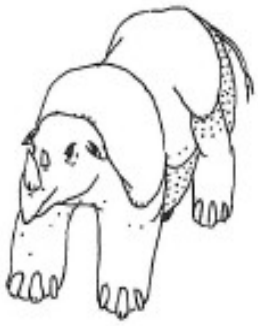
[VAN DER LINDEN 1993] Van der Linden, Elisabeth (1993). «Does Feedback Enhance Computer-Assisted Language Learning?» *Computers Education* 21 (1/2): 61–65.

Appendix A

Grammatisches Geschlecht

Alle Nomen haben ein grammatisches Geschlecht. Mit dem Begleiter kannst du das Geschlecht des Nomens erkennen. Das Nomen muss allerdings in der Einzahl stehen.

der Hund	männlich
die Frau	weiblich
das Haus	sächlich



14. Sortiere die Nomen nach dem grammatischen Geschlecht und schreibe sie in die richtige Spalte.

Affe, Kamel, Hund, Schlange, Blauwal, Meise, Schnecke, Krokodil, Schildkröte, Kuh, Pferd, Zebra, Steinbock, Delfin, Hase, Robbe, Schwein, Dromedar, Elefant, Biene, Nashorn

männlich	weiblich	sächlich

Singular und Plural von Nomen

Die meisten Nomen können in der Einzahl oder in der Mehrzahl stehen. Nun verwenden wir die Begriffe *Singular* für die Einzahl und *Plural* für die Mehrzahl.
Singular – Einzahl
Plural – Mehrzahl

Bei den meisten Nomen werden im Plural ein oder mehrere Buchstaben angehängt.
Beispiele: der Hund – die Hunde, das Kind – die Kinder, das Velo – die Velos

Bei einigen Nomen tönen der Singular und der Plural genau gleich.
Beispiele: der Spiegel – die Spiegel, der Teller – die Teller

Bei einigen Pluralformen entstehen Umlaute.
Beispiele: das Haus – die Häuser, das Schloss – die Schlösser

19. Suche die richtigen Pluralformen und schreibe sie auf.

Singular	Plural	Singular	Plural
die Mücke		der Löffel	
die Tanne		der Strauch	
das Dach		der Teppich	
das Schwert		die Gans	
das Loch		das Foto	
die Pfanne		der Schnauz	
der Deckel		der Turm	
das Schloss		der Traktor	
das Kissen		die Antwort	

Adjektive erkennen

Adjektive sagen dir, wie etwas ist oder wie etwas gemacht wird.
Das **kleine** Kind stolpert. Er lacht **laut** über diesen Witz.

Wenn du nicht sicher bist, ob ein Wort ein Adjektiv ist, kannst du es mit der **Einsetzprobe** überprüfen.

mutig → der mutige Knabe — richtig, also Adjektiv
sogar → der sogare Knabe — falsch, also kein Adjektiv

Die meisten Adjektive lassen sich steigern.
gross, grösser, am grössten

1. Überprüfe mit Hilfe der Einsetzprobe, welche Wörter durch den «Adjektivtrichter» fallen. Manchmal musst du die Wörter anpassen. Falls es richtig tönt, fallen sie durch den Trichter und du kannst sie unten aufschreiben.

fröhlich traurig hinzu lebendig
lustig sauber stumm daher rennend
bloss über bekannt feige abends
ruhig singend müde
blass immer blond krank

das ... Kind

Appendix B

Fragebogen Test-Stunde e-GramD
11. Juni 2007

Dein Name: _____

1. Welchen Text hast du ausgewählt (z.B. Verschiedenes, Fledermaus etc.)?

_____

2. Wie schwierig hast du deinen Text gefunden?

_____

3. Welche Übungsart hast du ausgewählt (Anklickübung, Lückentext, Puzzleübung)?

_____

4. Welche Übung hast du ausgewählt (z.B. Nomen, Nomen_Einzahl, Adjektive etc.)?

_____

5. Wie schwierig hast du es gefunden, deine Übung auszuwählen?

_____

6. Wie schwierig hast du deine Übung gefunden?

_____

7. Wie gut hast du Professor Würmlis Bewertung gefunden?

_____

8. Wie gut hast du Professor Würmlis Bewertung verstanden?

_____

9. Wie gut hast du Professor Würmlis Kommentare (Textblase über falschen Wörtern) gefunden?

_____

10. Wie gut hast du Professor Würmlis Kommentare (Textblase über falschen Wörtern) verstanden?

_____

11. Hat dir Professor Würmlis Kommentar (Textblase über falschen Wörtern) geholfen, die richtige Lösung zu finden?

_____

12. Was sollte Professor Würmli sonst noch können (andere Übungen, Texte, Komemntare, etc.)?

_____

Appendix C

Fragekatalog Lehrpersonen e-GramD

1. Welche Schulstufe unterrichtest du?

2. Ist die e-GramD ansprechend für deine Schüler?

3. Wie schwierig ist es für deine Schüler, links die richtige Übung auszuwählen?

4. Sind die Texte von e-GramD sinnvoll für deine Schüler?

5. Wie schwierig sind die Texte von e-GramD für deine Schüler?

6. Sind die Übungen von e-GramD sinnvoll für deine Schüler?

7. Wie schwierig sind die Übungen für deine Schüler?

8. Wie gut findest du die Bewertung der Übungen (Sprechblase von Professor Würmli)?

9. Wie gut findest du das Feedback von Professor Würmli (Sprechblase über falschen Wörtern/Eingaben)?

10. Würdest du diese Plattform im Unterricht gebrauchen?

11a. Wenn bei 7 ja: warum?

11b. Wenn bei 7 nein: was würdest du ändern?

12. Würdest du diese Plattform anderen Lehrern weiterempfehlen?

13a. Wenn bei 12 ja: warum?

13b. Wenn bei 12 nein: was müsste anders sein, damit du sie weiterempfiehlst?

Curriculum Vitae

Persönliche Informationen	Name: Sandra Lüthi Adresse: Albisstrasse 28, 8038 Zürich Telefon: +41 78 633 42 61 E-Mail: Sandra.Luethi@access.unizh.ch Geburtsdatum: 18. November 1978
Ausbildung	
2001–heute	Universität Zürich **Hauptfach Anglistik, 1. Nebenfach Computerlinguistik, 2. Nebenfach Soziologie**
1998–2000	Handelsschule KV, Lenzburg **Kaufmännische Berufsmatura**
1994–1998	Kantonsschule, Wohlen **Neusprachlicher Typus D**
Berufliche Tätigkeiten	
06.2004–05.2007	PersonalPoint GmbH, Küsnacht **Recruiting Assistant** (Teilzeit)
12.2005–04.2007	Englisches Seminar, Universität Zürich **Studierendenvertreterin Berufungskommission Linguistik**
09.2006–02.2007	Institut für Informatik, Computerlinguistik, Universität Zürich **Tutorin Vorlesung Formale Grammatiken und Syntaxanalysen**
03.2006–08.2006	Institut für Informatik, Computerlinguistik, Universität Zürich **Co-Tutorin Seminar Grammar Engineering**
10.2003–05.2004	The Purbeck School, Wareham, UK **Assistant German Teacher**
09.2000–08.2003	Fumedica AG, Muri AG **Sachbearbeiterin Kardiotechnik, Allgemeine Chirurgie und Export** (Teilzeit)
08. 1998–08.2000	Hotel Krone, Lenzburg **Kaufmännische Berufslehre mit BMA**

Hobbies	Badminton (Interclub BC Sporting) Sportklettern, Wandern Reisen
Ehrenamtliche Tätigkeiten	seit 2005 Treasurer Fachverein Anglistik
Sprachkenntnisse	Deutsch: Muttersprache Englisch: mündlich und schriftlich fliessend Französisch: mündlich sehr gut, schriftlich gut Italienisch: mündlich und schriftlich gut Schwedisch: mündlich und schriftlich Anfänger