Entwurf und Entwicklung eines GebärdenSchrift-Editors

Bachelorarbeit im Studiengang Informatik

Joachim Nitschke

Betreuer:

Dr. Guido Gryczan

Universität Hamburg
Fachbereich Informatik
Arbeitsbereich Softwaretechnik

Jun.-Prof. Dr. Barbara Hänel-Faulhaber

Universität Hamburg
Fachbereich Erziehungswissenschaft
Arbeitsbereich Behindertenpädagogik

Hamburg, den 23. Dezember 2009

Fachbereich Informatik

Universität Hamburg

Abstract: Gehörlose haben aufgrund des fehlenden Höreindrucks erhebliche Schwierigkeiten, die deutsche Schriftsprache zu erlernen. Um gehörlosen SchülerInnen und Erwachsenen die Schriftsprache nahe zu bringen, ist es erforderlich, den Zusammenhang zwischen Deutsch in deutscher Schriftsprache und der Deutschen Gebärdensprache (DGS) herzustellen. Da jedoch gehörlose Menschen das Deutsche auditiv nicht (vollständig) wahrnehmen können, soll der Einsatz von lautsprachbegleitenden Gebärden (LBG), dargestellt mit Hilfe der GebärdenSchrift, als Brücke zwischen DGS und der deutschen Schriftsprache fungieren. Eine adäquate IT-basierte Unterstützung für die Erstellung von Unterrichtsmaterialien in GebärdenSchrift existiert nicht. Mit Hilfe Agiler Methoden wird im Rahmen dieser Arbeit ein Prototyp für einen GebärdenSchrift-Editor entwickelt, der vor allem das schnelle Übersetzen in LBG unterstützt. Der GebärdenSchrift-Editor wird als browserfähige Online-Anwendung auf Basis des Google Web Toolkits realisiert. Diese Arbeit führt zunächst in die Thematik der Gebärdensprache und -Schrift ein und beschreibt anschließend die Entwicklung des GebärdenSchrift-Editors aus softwaretechnischer Sicht.

Inhaltsverzeichnis

1 Einleitung und Motivation **5**

1.1 Zur Entstehung dieser Arbeit 6

2 Fachlicher Hintergrund **7**

2.1 Die deutsche Gebärdensprache 7

2.1.1 Phonologie 8

2.1.2 Morphologie 10

2.1.3 Syntax 12

2.2 Lautsprachbegleitende Gebärden 13

2.3 SignWriting/GebärdenSchrift 14

2.3.1 Symbole 16

2.3.2 Das International SignWriting Alphabet 19

2.3.3 Andere Notationsformen für Gebärden 21

2.4 Probleme an Gehörlosenschulen/Anwendungskontext 22

3 Anforderungsanalyse **25**

3.1 Anforderungen 25

3.2 Aktuelle GebärdenSchrift-Software 29

3.2.1 SignWriter DOS 30

3.2.2 SignWriter Java 32

3.2.3 SignPuddle Online 33

3.2.4 Fazit 36

4 Entwurf des GebärdenSchrift-Editors **37**

4.1 Der Übersetzungsmodus 38

4.2 Der Bearbeitungsmodus 39

4.3 Einfügen von GebärdenSchrift-Symbolen mit dem Tastaturfeld 40

5 Softwaretechnische Umsetzung **42**

5.1 Verwendete Methode 42

5.2 Verwendete Technologien 43

5.2.1 Das Google Web Toolkit 43

5.3 Architektur 44

5.3.1 Material 45
5.3.2 Werkzeug 47
5.3.3 Webservice 50

6 Fazit und Ausblick **52**

A Planung **57**

B Mock-Up **59**

1 Einleitung und Motivation

Aufgrund des fehlenden Höreindrucks haben Gehörlose erhebliche Schwierigkeiten, die deutsche Schriftsprache zu erlernen. Denn die deutsche Schriftsprache ist eine Verschriftlichung der deutschen (Laut-)Sprache. Ich gehe in dieser Arbeit jedoch davon aus, dass Gehörlose vermehrt in der deutschen Gebärdensprache (DGS) kommunizieren. Sie hat eine ganz eigene Grammatik und viele Konzepte, die in der Lautsprache nicht existieren. Ebenso existieren in der Lautsprache Konzepte, die es in der Gebärdensprache nicht gibt, zum Beispiel Deklination oder Konjugation. Das Erlernen der deutschen Schriftsprache ist (für Hörende) also vergleichbar mit dem Erlernen einer Fremdsprache mit unterschiedlichen Schriftsymbolen (etwa Deutsch/Chinesisch). Allerdings können Gehörlose dabei die Buchstaben nicht mit Lauten verknüpfen, was das Lernen zusätzlich erschwert. Während Hörende ein geschriebenes Wort auf seine Aussprache abbilden können, müssen Gehörlose die Schreibweise des Wortes auswendig lernen. Ein Bezug zwischen DGS und der deutschen Lautsprache bzw. Schriftsprache existiert höchstens über die Mundbilder, die teilweise beim Gebärden verwendet werden.

Aus den zuvor genannten Gründen beherrschen viele Gehörlose die deutsche Schriftsprache nicht gut. Ihnen die Schriftsprache zu vermitteln, ist deshalb ein zentrales Anliegen der schulischen Ausbildung sowie der Erwachsenenbildung. Um gehörlosen SchülerInnen und Erwachsenen die Schriftsprache nahe zu bringen, ist es erforderlich, den Zusammenhang zwischen Deutsch in deutscher Schriftsprache und der DGS herzustellen. Dabei sollen lautsprachbegleitenden Gebärden in verschriftlichter Form als Mittlerfunktion zwischen DGS und der deutschen Lautsprache beim Erlernen und Kontrastieren der deutschen Schriftsprache zum Einsatz kommen. Denn die lautsprachbegleitende Gebärdensprache (LBG) verbindet die Grammatik der deutschen Schriftsprache mit den Gebärden der DGS. Zur Erstellung von entsprechendem Unterrichtsmaterial ist eine Vergegenständlichung bzw. Verschriftlichung der DGS und der LBG notwendig, um diese untereinander und mit der deutschen Schriftsprache in Beziehung zu setzen. Dabei kommt in diesem Fall die sehr ikonische (bildhafte) GebärdenSchrift zum Einsatz.

Für die Erstellung von Unterrichtsmaterialien in GebärdenSchrift existiert bisher keine adäquate IT-basierte Unterstützung. Aus diesem Grund ist für Lehrende an Gehörlosenschulen die Hürde sehr groß, diese Materialien anzufertigen. Aufgrund besonderer Eigenschaften der Gebärdensprache und der GebärdenSchrift gelten für das Erstellen von GebärdenSchrift-Texten andere Anforderungen als für das Erstellen von Texten in der deutschen Schriftsprache bzw. in lateinischer Schrift. Ziel dieser Arbeit ist es deshalb, einen Prototypen für einen GebärdenSchrift-Editor zu entwickeln, mit dessen Hilfe sich GebärdenSchrift-Texte ähnlich schnell und anschaulich erstellen lassen, wie mit gängigen Textverarbeitungsprogrammen.

Diese Arbeit führt zunächst in die Thematik der Gebärdensprache und -Schrift ein und beschreibt anschließend die Entwicklung des GebärdenSchrift-Editors aus softwaretechnischer Sicht.

Im folgenden Kapitel beschreibe ich zunächst den fachlichen Hintergrund. Es handelt sich dabei um Hintergrundwissen, das für das tiefere Verständnis der Problemstellung notwen-

dig ist. Zunächst erläutere ich die Charakteristika der Deutschen Gebärdensprache und der Lautbegleitenden Gebärdensprache und deren Unterschiede vor dem Hintergrund der deutschen Lautsprache. Anschließend gebe ich eine Einführung in die in der entwickelten Anwendung verwendeten GebärdenSchrift. Am Ende des zweiten Kapitels wird die in der Einleitung beschriebene Problemstellung und der Anwendungskontext ausführlicher erläutert.

Im dritten Kapitel werden aus den im zweiten Kapitel beschriebenen Sachverhalten Anforderungen für einen GebärdenSchrift-Editor abgeleitet. Ich stelle zunächst eine Liste mit begründeten Anforderungen zusammen. Vor dem Hintergrund dieser Anforderungen folgt eine Diskussion aktueller Anwendungen im Zusammenhang mit der GebärdenSchrift.

Die so gewonnenen Erkenntnisse sind die Grundlage für den Entwurf des im Rahmen dieser Arbeit entwickelten GebärdenSchrift-Editors. Dieser wird im vierten Kapitel vorgestellt. Dabei beschreibe ich, auf welche Weise die Anforderungen aus dem vorherigen Kapitel umgesetzt wurden und motiviere ausführlich jeweils getroffene Designentscheidungen.

Das fünfte Kapitel dieser Arbeit beschäftigt sich mit der softwaretechnischen Umsetzung des zuvor beschriebenen Entwurfes. Ich beschreibe kurz die verwendete Methode bei der Projektplanung und stelle das Google Web Toolkit als zentrale verwendete Technologie vor. Anschließend beschreibe ich die zu Grunde liegende Software-Architektur.

Das letzte Kapitel fasst die Arbeit zusammen und bietet einen Ausblick: Welche Aspekte des Entwurfs wurden realisiert und welche nicht? Inwiefern lässt sich der Entwurf erweitern?

1.1 Zur Entstehung dieser Arbeit

Ausgangssituation dieser Arbeit war das studentische Projekt „Objektorientierte Softwareentwicklung“ am Fachbereich Informatik der Universität Hamburg. Ich lerne seit einem Jahr aus persönlichem Interesse die deutsche Gebärdensprache. Als ich erfuhr, dass sich das zuvor genannte Projekt mit der in der Einleitung beschriebenen Thematik beschäftigt, habe ich mich entschlossen, an dem Projekt teilzunehmen und meine Bachelorarbeit über das Thema zu verfassen.

Der Autor des Buches „Handbuch zur GebärdenSchrift“ (Wöh05) Stefan Wöhrmann spielt innerhalb dieses Projektes eine wichtige Rolle. Er ist Diplompsychologe und arbeitet als Lehrer an der Gehörlosenschule in Osnabrück (Landesbildungszentrum für Hörgeschädigte). Er hat die GebärdenSchrift für den deutschen Sprachgebrauch angepasst und arbeitet mit ihr erfolgreich in seinem Unterricht, was ihn zu dem deutschen Experten auf diesem Gebiet macht.

Die Erfahrungen von Stefan Wöhrmann im Umgang mit teilweise veralteter GebärdenSchrift-Software gaben den Anlass zur Entwicklung einer zeitgemäßen, IT-basierten Unterstützung für seinen Unterricht. Bei einer Hospitation in seinem Unterricht und im Laufe mehrerer Interviews hatte ich die Gelegenheit, einen ausgiebigen Einblick in diesen

Anwendungskontext zu erlangen. Die so gewonnenen Erkenntnisse sind unter anderem Bestandteil des folgenden Kapitels.

2 Fachlicher Hintergrund

Dieses Kapitel erläutert die Problemstellung und den Anwendungskontext genauer. Dazu beschreibe ich zunächst Wissen, das im weiteren Verlauf der Arbeit vorausgesetzt wird. So erläutere ich zunächst die Besonderheiten der deutschen Gebärdensprache.

2.1 Die deutsche Gebärdensprache

Die deutsche Gebärdensprache (DGS) ist eine eigenständige Sprache. Sie unterscheidet sich vor allem in den Modalitäten offensichtlich grundlegend von der deutschen Lautsprache. Während die deutsche Lautsprache mit Hilfe des Ohres und der Sprechorgane funktioniert, verwendet die DGS den Körper als Sprachinstrument. Die Rezeption der Sprache erfolgt visuell. Dies bietet Gehörlosen die Möglichkeit, genauso wie Hörende miteinander zu kommunizieren. Bei der Kommunikation in Gebärdensprache steht die manuelle Komponente, also die mit der Hand ausgeführten Zeichen, im Vordergrund. Zusätzlich zur manuellen Komponente werden auch Mimik, Kopf- und Körperhaltung zum Ausdruck grammatikalischer Merkmale eingesetzt. Ein weiterer Unterschied zur Lautsprache ist weniger offensichtlich: Die DGS unterliegt im Hinblick auf die Grammatik anderen Regeln und Prinzipien. In diesem Abschnitt erläutere ich speziell diese grammatikalischen Unterschiede. (Vgl. BKPZ06, S. 22)

Ein weiterer Unterschied zur deutschen Lautsprache besteht darin, dass innerhalb der DGS größere regionale Unterschiede existieren. Je nach Region werden unterschiedliche Gebärden verwendet (vgl. Fri07). Im Gegensatz zur deutschen Lautsprache hat sich noch kein allgemeines Hochdeutsch herausgebildet. Dies liegt unter anderem daran, dass die Gebärdensprache lange Zeit nicht offiziell als Sprache anerkannt wurde (vgl. Abschnitt 2.4).

Ein allgemeines Verständnis der Gebärdensprache ist zum einen Voraussetzung, um die Problemstellung genauer zu verstehen, denn gerade die zuvor genannten grammatikalischen Unterschiede zwischen der DGS und der Lautsprache bereiten Gehörlosen Probleme beim Erlernen der deutschen Schriftsprache. Zum anderen resultieren aus den Besonderheiten der DGS spezielle Anforderungen an die Erstellung von DGS-Texten in der GebärdenSchrift, also auch für die Erstellung eines entsprechenden Editors.

Ich orientiere mich bei der Vorstellung der DGS-Grammatik an dem Buch „Grammatik der Deutschen Gebärdensprache“ (PvMMH08). Gemäß den Teilgebieten der Linguistik unterteilt dieses Buch die Betrachtung der DGS in drei Abschnitte: die Phonologie, die Morphologie und die Syntax. Die Bedeutung der Teilgebiete wird jeweils zu Beginn der hier folgenden Abschnitte erläutert. Ich lege den Schwerpunkt jeweils auf diejenigen Aspekte der DGS-Grammatik, die sie von der lautsprachlichen Grammatik unterscheiden.

2.1.1 Phonologie

Die Phonologie ist wörtlich die „Lehre von den (Sprech-)Lauten“. Im Allgemeinen bezeichnet sie jedoch die Lehre von den kleinsten bedeutungsunterscheidenden Elementen einer Sprache (Phoneme) und wird daher auch auf die Gebärdensprache übertragen. Ein kleinstes bedeutungsunterscheidendes Element ist das kleinste Element, dessen Änderung gerade noch die Bedeutung der sprachlichen Einheit verändert, zu der es gehört. Da die Phoneme der deutschen Lautsprache die Laute sind, ist ihr Name wie eingangs erwähnt von dem Wort „Phon“ für „Sprechlaut“ abgeleitet. So führt zum Beispiel die Änderung des ersten Lautes in dem Wort „Haus“ zu den bedeutungsverschiedenen Wörtern „Maus“, „Laus“ oder „raus“. Ebenso kann auch die Änderung eines der im Folgenden beschriebenen Phoneme der DGS zu Gebärden mit anderen Bedeutungen führen.

Zweck dieses Abschnitts ist es vor allem, zu beschreiben, welche Mittel einem Gebärdenden beim Gebärden zur Verfügung stehen, um verschiedene Bedeutungen auszudrücken.

Die Phoneme von Gebärdensprachen lassen sich wie zuvor bereits angedeutet in manuelle und nonmanuelle Komponenten aufteilen. Während erstere sich auf die Hand beziehen, beinhalten letztere zum Beispiel die Mimik oder die Kopfhaltung.

Eine *manuelle Komponente* gehört zu jeder Gebärde. Sie ist ihr zentrales Element. Manuelle Anteile einer Gebärde lassen sich anhand folgender vier Parameter unterscheiden:

- *Handform*: Unter der Handform versteht man die äußere Gestalt der Hand. Mögliche Handformen sind zum Beispiel die Faust, die flache Hand oder ein ausgestreckter Zeigefinger. Es gibt zahlreiche verschiedene Handformen.

- *Handstellung*: Die Handstellung beschreibt ihre Orientierung im Raum, speziell die der Handfläche und des Fingeransatzes (Verlängerung des Handrückens). Zum Beispiel kann die Handkante, die Handfläche oder der Handrücken dem Betrachter zugewandt sein. Ein Zeigefinger kann nach hinten, nach oben, nach unten, nach unten links usw. zeigen.

- *Ausführungsstelle*: Die manuelle Komponente einer Gebärde kann an verschiedenen Orten ausgeführt werden. Diese Orte können einerseits im Raum um den Gebärdenden liegen und andererseits an seinem Körper. Beispiele wären an der Nasenspitze, am Bauch oder vor dem Gebärdenden auf Kopfhöhe.

- *Bewegung*: Dieser Parameter umfasst, wie und wohin die Hand bewegt wird. Bewegungen lassen sich in handexterne (Bewegungen im Gebärdenraum, also im Raum um den Gebärdenden) und handinterne Bewegungen (zum Beispiel Bewegungen der Finger) unterteilen. Während die ersten drei Parameter in jeder Gebärde vorkommen, beinhaltet nicht jede Gebärde eine Bewegung.

Vollziehen beide Hände in einer Gebärde die gleiche Bewegung, spricht man von einer symmetrischen, vollziehen sie unterschiedliche Bewegungen dementsprechend von einer nichtsymmetrischen Gebärde. In letzterem Fall ist immer eine der beiden Hände konsequent dominant. Das heißt, diese Hand vollzieht die Hauptbewegung, während die andere

nur assistiert oder nicht beteiligt ist. Bei Rechtshändern ist in der Regel die rechte Hand die dominante, bei Linkshändern entsprechend die linke.

In der *nonmanuellen Komponente* einer Gebärde sind alle Sprachmittel zusammengefasst, bei denen die Hand **nicht** zum Einsatz kommt. Dabei gibt es vier zentrale Elemente:

- *Mimik*: Die Mimik hat in der DGS neben dem Ausdruck von Emotionen auch eine sprachliche Funktion. Sie kann einerseits die Bedeutung einer Gebärde illustrieren. Die Gebärde für „grimmig“ wird zum Beispiel durch einen entsprechenden Gesichtsausdruck begleitet. Andererseits kommt die Mimik auf syntaktischer Ebene einer adverbialen Bestimmung gleich. Durch das simultane Begleiten von Gebärden mit entsprechender Mimik, wird eine zusätzliche Information übermittelt – so zum Beispiel der Unterschied zwischen „motiviert arbeiten“ und „gelangweilt arbeiten“. Darüber hinaus spielt die Mimik auch bei der Morphologie und der Syntax eine Rolle.

- *Mundbewegungen*: Die Mundbewegungen werden unterteilt in Wortbild und Mundgestik. Unter Wortbildern versteht man das Nachformen der lautsprachlichen Sprechbewegungen mit dem Mund simultan zu den meisten Gebärden. Die Wortbilder sind durch die sehr lautsprachlich geprägte Gehörlosenbildung in Deutschland entstanden. Mundgestik hingegen bezeichnet die Mundbilder, die keinen Bezug zur Lautsprache haben. Zum Beispiel formt ein Gebärdender simultan zur Gebärde für „besitzen“ den Mund so, als würde er einen „Sch“-Laut aussprechen (siehe Abbildung 1).

Abbildung 1: Die Gebärde für „besitzen“ wird durch die zu einem „Sch“-Laut passende Mundgestik begleitet. (Quelle: BKPZ06, S. 98)

- *Blickrichtung*: Die Blickrichtung spielt bei der Kommunikation in Gebärdensprache eine noch wichtigere Rolle als bei der lautsprachlichen Kommunikation. Sie lenkt die Aufmerksamkeit des Kommunikationspartners. Der Blick zum Gesprächspartner symbolisiert Gesprächsbereitschaft. Der Blick in den Raum dient als Hinweis oder als Rückverweis auf zuvor verortete Konzepte (siehe Abschnitt 2.1.2).

- *Kopf- und Oberkörperhaltung*: Die Kopf- und Oberkörperhaltung gehört größtenteils zum persönlichen Ausdruck des Gebärdenden und übernimmt nur manchmal eine sprachliche Funktion. Ein Kopfnicken bzw. Kopfschütteln drückt zum Beispiel Zustimmung bzw. Ablehnung aus. Ebenso werden durch weitere Kopfhaltungen andere Sachverhalte ausgedrückt. Eine wichtige Rolle spielt die Körperhaltung bei der Rollenübernahme (siehe Abschnitt 2.1.3).

Der folgende Abschnitt beschäftigt sich mit dem zweiten der zuvor genannten Teilgebeite der Linguistik: der Morphologie.

2.1.2 Morphologie

Neben der Phonologie ist ein weiterer Bereich der Linguistik die Morphologie. Sie ist die Lehre von den kleinsten bedeutungstragenden Elementen einer Sprache (Morpheme) und beschäftigt sich in der Lautsprachlinguistik mit der inneren Struktur von Wörtern. Dazu gehören einerseits Regeln zur Wortbildung, also der Kombination mehrerer Morpheme zu einem Wort. Ein einfaches Beispiel dafür ist die Bildung von Komposita. Dies sind zusammengesetzte Wörter wie „Donaudampfschifffahrtsgesellschaft“. Andererseits zählt auch die Flexion von Wörtern zur lautsprachlichen Morphologie. Die Flexion oder Beugung eines Wortes bezeichnet seine Anpassung zum Ausdruck einer grammatikalischen Funktion. Beispiele sind die Konjugation von Verben oder die Deklination von Nomen.

In der DGS existieren ebenfalls Regeln zur Bildung von Gebärden und zu ihrer Flexion. Diese unterscheiden sich jedoch grundlegend von denen der Lautsprache. In diesem Abschnitt wird besonders deutlich, inwiefern sich die DGS von der deutschen Lautsprache unterscheidet. Ich stelle zunächst zwei Prinzipien der Gebärdenbildung vor: die *Bildhaftigkeit (Ikonizität)* und die *Inkorporation*.

Gebärdensprachen sind visuell. Die manuelle Komponente vieler Gebärden projiziert sprachliche Äußerungen in den Gebärdenraum, indem zum Beispiel eine Tätigkeit imitiert oder die Form eines Gegenstandes nachgebildet wird. Ihre *Bildhaftigkeit (Ikonizität)* ist deshalb eines der deutlichsten Merkmale von Gebärdensprachen. Der bildhafte (ikonische) Anteil einer Gebärde kann mehr oder weniger stark ausgeprägt sein. Beispielsweise ist die DGS-Gebärde für „schwimmen“ sehr ikonisch: Der Gebärdende imitiert mit seinen Armen die Schwimmbewegung (siehe Abbildung 2). Die Hand übernimmt dabei in Gebärden verschiedene Funktionen. Bei der Gebärde für „schwimmen“ symbolisiert sie sich selbst (Manipulator). In anderen Gebärden symbolisiert die Hand zum Beispiel durch eine spezifische Handform einen Gegenstand (Substitutor). Alternativ kann sie auch als Zeichenwerkzeug dienen, um das Erscheinungsbild von Dingen in den Gebärdenraum zu zeichnen (Skizze), als Hinweispfeil, um auf etwas zu zeigen (Index), oder als Maß, um die Ausdehnung bzw. die Größe von etwas anzuzeigen (Maß).

Im Gegensatz zu Lautsprachen sind Gebärdensprachen nicht zwangsläufig sequenziell. Sie können auch mehrere Informationen simultan vermitteln. Neben der Parallelisierung von manuellen und nonmanuellen Komponenten ist ein Mittel dazu die *Inkorporation*. Sie ist ein weiteres Prinzip zur Gebärdenbildung und bezeichnet die Vereinigung zweier Gebärden und ihrer Bedeutungen. Sie geschieht zum Beispiel im Zusammenhang mit Zahlen.

Abbildung 2: Die Gebärde für „schwimmen“. (Quelle: BKPZ07, S. 148)

Die Gebärde für „vier“ ist eine Hand mit vier ausgestreckten Fingern. Wird die Gebärde für „Stunde“ mit vier statt einem ausgestreckten Finger ausgeführt, steht sie für „vier Stunden“. Ebenso kann die Verneinung mit Hilfe einer alphaförmigen Bewegung in bestimmte Gebärden inkorporiert werden. Ein Beispiel dafür ist die Gebärde für „nicht stimmen“. Sie besteht zum einen aus der alphaförmigen Bewegung für die Verneinung und zum anderen aus der Handform aus der Gebärde für „stimmen“ (siehe Abbildung 3).

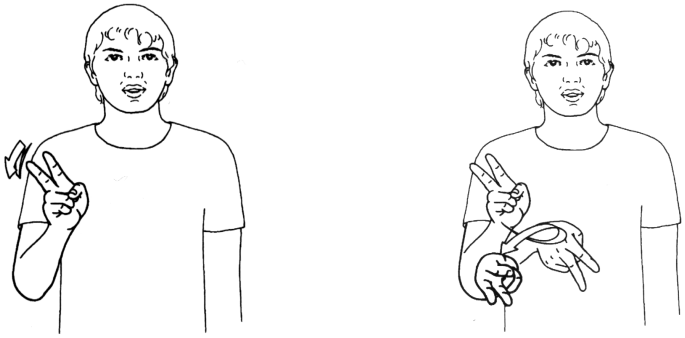


Abbildung 3: Die alphaförmige Bewegung bei der Inkorporation der Verneinung: Die Gebärden für „stimmen“ (links) und „nicht stimmen“ (rechts). (Quelle: BKPZ07, S. 44/130)

Wie zu Beginn dieses Abschnitts erwähnt zählen auch Regeln der Flexion zum Gebiet der Morphologie. Die Flexion erfolgt in der DGS auf andere Weise als in der deutschen Lautsprache. Verben richten sich teilweise nicht nur nach dem Subjekt (Subjektkongruenz), sondern anders als in der deutschen Lautsprache auch nach dem Objekt (Objektkongruenz). Eine Anpassung an die Zeit erfolgt nicht. Des Weiteren werden Nomen in der DGS nicht flektiert. Die Einteilung in die klassischen Wortartkategorien „Nomen“, „Adjektiv“, „Verb“ usw. erfolgt dabei in der DGS nicht rein formal sondern kontextbezogen. Die Gebärde für „Arbeit“ gleicht zum Beispiel der Gebärde für „arbeiten“. Ob es sich um das Verb oder das Nomen handelt, wird nur im Satzzusammenhang deutlich.

Abbildung 4: Drei Beispiele für Transferverben: die Gebärden für „schicken“, „geben“ und „zeigen“ (von links nach rechts). Die verschiedenen Pfeile beim rechten Beispiel deuten an, wie die Bewegungsrichtung verändert werden kann. (Quelle: BKPZ07, S. 69)

Um zu erläutern, wie die Flexion in der DGS funktioniert, beschreibe ich im Folgenden beispielhaft eine Unterkategorie der Verben und deren Anpassung an unterschiedliche Handlungsrollen im Satz: die *Transferverben* oder *Richtungsverben*. Transferverben beschreiben den Übergang eines Gegenstandes oder einer zwischenmenschlichen Botschaft von einer Quelle zu einem Ziel. Ein Gegenstand wird zum Beispiel „übergeben“ oder „verschickt“. Eine Botschaft wird beispielsweise beim „Fragen“ in Form der gestellten Frage übermittelt. Abbildung 4 zeigt einige Beispiele. In den Gebärden „wandert“ dann das Objekt oder die Botschaft, repräsentiert durch die Hand, von der Quelle zum Ziel. Dabei sind in der Verbform Subjekt und Objekt enthalten. Das Subjekt wird durch den Anfangspunkt der Bewegung repräsentiert, das Objekt durch den Endpunkt. Die Flexion erfolgt hier durch eine Änderung der Bewegungsrichtung. Es gibt allerdings auch Transferverben, die keine Bewegung beinhalten, sondern nur ihre Handstellung auf die zuvor erwähnten Punkte ausrichten.

Die hier beschriebenen Mechanismen sind dafür verantwortlich, dass GebärdenSchrift-Texte nicht nur aus vordefinierten Gebärden zusammengesetzt werden können, obwohl dies dadurch begünstigt wird, dass viele Gebärden nicht flektiert werden. Dies wird bei der Anforderungsanalyse in Abschnitt 3.1 genauer beschrieben. Der folgende Abschnitt beschäftigt sich mit dem dritten der zuvor genannten Teilgebiete der Lingusitik: der Syntax in der DGS.

2.1.3 Syntax

Die Syntax beschreibt in der Linguistik, aus welchen Bestandteilen Sätze bestehen und auf welche Weise diese kombiniert werden. Ebenso wie in der deutschen Lautsprache lassen sich in der deutschen Gebärdensprache die Satzglieder Subjekt, Objekt und Prädikat unterscheiden. In einem klassischen DGS-Satz werden die Satzglieder durch einzelne Gebärden repräsentiert und stehen in dieser Reihenfolge. Dies ist jedoch nicht immer so. Im vorherigen Abschnitt habe ich bereits beschrieben, wie bei den Transferverben Subjekt und Objekt mit dem Prädikat zu einer Gebärde verbunden werden. Des Weiteren kann ein

DGS-Satz auch ohne ein Verb gebildet werden. Nomen übernehmen dann die Funktion des Prädikats. Die Aussage „Dort stehen zwei Häuser.“ kann allein durch das zweimalige Gebärden der Gebärde für „Haus“ nebeneinander ausgedrückt werden. Die beiden Häuser werden in diesem Beispiel verortet.

Der Begriff *Verortung* bezeichnet ein besonderes sprachliches Mittel der DGS, das globalen Variablen ähnelt. Ein Gebärdender kann Konzepte wie Personen oder Gegenstände im Raum verorten. Das heißt, er weist ihnen eine spezifische Stelle im Raum zu. Im weiteren Gesprächsverlauf kann er durch Zeigen an diese Stelle Bezug auf das dort verortete Konzept nehmen. Ebenso können inhaltliche oder räumliche Beziehungen zwischen ihnen vermittelt werden – je nachdem, ob es sich bei den verorteten Konzepten um Lebewesen, Gegenstände oder Orte handelt.

Ein weiteres besonderes sprachliches Mittel der Gebärdensprachen ist die *Rollenübernahme*. Um an einer Szene beteiligte Lebewesen zu beschreiben, kann ein Gebärdender ihre Rolle übernehmen. Bei der Rollenübernahme versetzt sich der Gebärdende in das dargestellte Lebewesen hinein. Dabei ist besonders die Imitation von nonmauellen Komponenten, wie Mimik oder Körperhaltung, sowie die richtige Orientierung zu anderen dargestellte Lebewesen entscheidend. Hat der Gebärdende dadurch die Rolle des betreffenden Lebewesens übernommen, kann er aus der Rolle heraus dessen Aktionen beschreiben.

In diesem Abschnitt habe ich erläutert, dass die DGS nicht einfach eine andere Repräsentation der deutschen Lautsprache ist, sondern eine eigenständige Sprache mit einer unterschiedlichen Grammatik. Dennoch können Gebärden auch genutzt werden, um die lautsprache visuell darzustellen. In diesem Fall spricht man von lautsprachbegleitenden Gebärden.

2.2 Lautsprachbegleitende Gebärden

Die lautsprachbegleitende Gebärdensprache (LBG) verwendet Gebärden, um einen lautsprachlichen Satz auszudrücken. Das heißt, ein Satz wird syntaktisch nahezu genauso gebärdet, wie er in der jeweiligen Lautsprache gesprochen wird. Dazu ist es teilweise notwendig, künstliche Gebärden einzuführen, um Wörter zu übersetzen, deren Bedeutung in der Gebärdensprache nicht durch Gebärden sondern auf andere Weise ausgedrückt werden. Dies gilt zum Beispiel für Artikel, bestimmte Konjunktionen oder Adverbien.

Im Alltag wird die LBG meistens von Gehörlosen oder Schwerhörigen genutzt, die ihr Gehör erst in höherem Alter verloren haben und somit noch die jeweilige Laut- und Schriftsprache gelernt haben, nicht aber die DGS und ihre Grammatik. Von Geburt an Gehörlose nutzen die LBG in der Regel nicht. Denn bei der Alltagskommunikation ist das Gebärden des kompletten lautsprachlichen Satzes ein Nachteil. Die LBG erbt damit die Schwächen beider Sprachen: Bewegungen mit der Hand zu vollziehen dauert länger und ist physisch anstrengender als das Produzieren von Lauten mit dem Sprechapparat. Die Gebärdensprache gleicht diesen Nachteil durch ihre Kompaktheit aus. Durch Ausnutzung des Raumes kann, wie bereits erwähnt, auf viele Füllwörter verzichtet werden. Sprachliche

Mittel wie die in Abschnitt 2.1.2 beschriebene Inkorporation erlauben es zudem, mehrere Sachverhalte simultan auszudrücken. In der LBG gehen diese Vorteile verloren.

Im in dieser Arbeit beschriebenen Lernkontext hat die LBG jedoch einen entscheidenden Vorteil: Da die LBG die Gebärden der DGS mit der Grammatik der deutschen Lautsprache verbindet, kann sie als Bindeglied zwischen den beiden Sprachen angesehen werden. Diese Eigenschaft kann im bilingualen Unterricht (vgl. Abschnitt 2.4) an Gehörlosenschulen dafür genutzt werden, Gehörlose beim Erlernen der deutschen Schriftsprache zu unterstützen. Durch eine Gegenüberstellung eines lautsprachlichen Satzes und des gleichen Satzes in LBG werden die Worte und Zeichen in lateinischer Schrift mit den Gebärden verknüpft. Vor allem aber wird so die Grammatik der deutschen Laut- und Schriftsprache mit den Mitteln der Gebärdensprache dargestellt. Dies bedeutet jedoch nicht, dass ein Gehörloser, der die DGS beherrscht, einen in LBG repräsentierten lautsprachlichen Satz automatisch versteht. Wenngleich er einige Gebärden versteht, wird er dennoch erst die neue Grammatik und die zuvor erwähnten zusätzlichen Gebärden erlernen müssen.

Die Übersetzung eines lautsprachlichen Satzes in LBG lässt sich aufgrund der direkten Abbildung der einzelnen Wörter auf Gebärden zum Teil automatisieren. Anders als bei der Übersetzung in DGS kann jedem Wort eine LBG-Gebärde zugewiesen werden. Dennoch existieren Ausnahmen dieser Regel, die eine vollständig automatisierte Übersetzung stark verkomplizieren. Beispielsweise können für ein Wort mehrere synonyme Übersetzungen existieren, von denen nicht alle die im Kontext korrekte Bedeutung repräsentieren. Abbildung 5 zeigt ein Beispiel: Die „Bank“ als Geldinstitut wird anders gebärdet als die „Bank“ als Sitzgelegenheit.

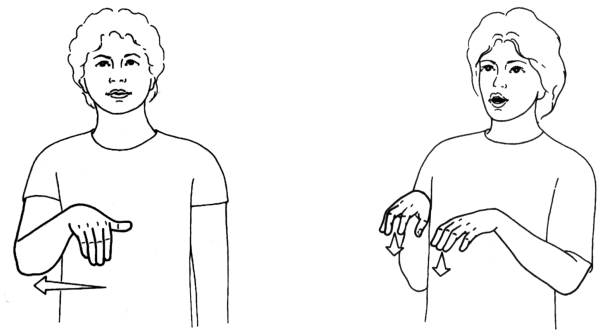


Abbildung 5: Zwei Gebärden mit der Bedeutung „Bank“: links die Sitzgelegenheit, rechts das Geldinstitut. (Quelle: BKPZ07, S. 43/120)

2.3 SignWriting/GebärdenSchrift

Die GebärdenSchrift wird in der im Rahmen dieser Arbeit entwickelten Anwendung zur Verschriftlichung der DGS und der LBG verwendet. Die Rolle der GebärdenSchrift im Unterricht Gehörloser wird in Abschnitt 2.4 noch genauer beschrieben.

Die GebärdenSchrift entstand aus einem allgemeinen System zur Beschreibung von Bewegungen, das von der amerikanischen Tänzerin Valerie Sutton zunächst zur Erfassung von Tanzchoreographien erfunden wurde. Aus dem *Sutton Movement Writing* entwickelte sie anschließend ein Notationssystem für Gebärden: das *Sutton SignWriting*. Der in Abschnitt 1.1 vorgestellte deutsche Diplom-Psychologe und Lehrer an der Gehörlosenschule Osnabrück Stefan Wöhrmann übernahm und erweiterte das Sutton SignWriting für seinen Unterricht. Auf diese Weise entstand eine deutsche Variante, die *GebärdenSchrift*. Die deutsche Schreibweise mit einem großen „S“ in der Mitte folgt aus der amerikanischen Schreibweise von „SignWriting“. Stefan Wöhrmann ist Autor des Buches „Handbuch zur GebärdenSchrift“ (Wöh05), das als Literaturquelle für diesen Abschnitt dient.

Die GebärdenSchrift stellt die Phoneme der Gebärdensprache über Symbole dar, die ich in nächsten Abschnitt genauer erläutere. Im Gegensatz zu vielen anderen Schriften werden diese Symbole jedoch nicht linear, sondern zweidimensional zu einer verschriftlichten Gebärde kombiniert (siehe Abbildung 6). Auf diese Weise können die räumlichen Gebärden anschaulicher dargestellt werden. Die verschriftlichten Gebärden werden dann entweder von links nach rechts oder von oben nach unten linear angeordnet, um Texte zu bilden. Aufgrund des Bezugs zur deutschen Schriftsprache priorisiere ich in dieser Arbeit die Schreibweise von links nach rechts. In der zweidimensionalen Anordnung der Symbole zur Bildung von Gebärden liegt die Besonderheit dieser Schrift im Vergleich zum Beispiel zur lateinischen Schrift. Sie ist auch dafür verantwortlich, dass Texte in GebärdenSchrift nicht einfach mit einem anderen Zeichensatz in klassischen Textverarbeitungsprogrammen erstellt werden können. Denn diese reihen die Symbole linear aneinander.

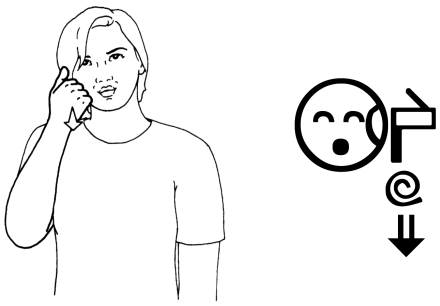


Abbildung 6: Die Gebärden für „bitte“ repräsentiert in der GebärdenSchrift (rechts). (Quelle: BKPZ06, S. 38)

Die GebärdenSchrift beschreibt die Gebärden aus der *expressiven Sichtweise*. Das heißt, die Ausrichtung und Anordnung zum Beispiel der Hände wird aus der Perspektive des Gebärdenden dargestellt, bzw. von einem Standpunkt hinter dem Gebärdenden. Dementsprechend erscheint die rechte Hand auf der rechten Seite, die linke Hand analog auf der linken Seite.

Im Folgenden stelle ich die Symbole der GebärdenSchrift vor. Dabei orientiere ich mich an den in Abschnitt 2.1.1 beschriebenen Phonemen der DGS und beschreibe jeweils, wie diese in der GebärdenSchrift abgebildet werden.

2.3.1 Symbole

Die GebärdenSchrift besteht aus einer Menge von Symbolen, welche die in Abschnitt 2.1.1 genannten Phoneme von Gebärdensprachen abbilden. Da es in Gebärdensprachen sehr viele Phoneme gibt, ist die Menge der GebärdenSchrift-Symbole verglichen mit der lateinischen Schrift extrem groß. Ich stelle im Folgenden einige grundlegende Symbole vor, um einen Einblick in die Funktionsweise der GebärdenSchrift zu geben. Aufgrund der großen Symbolvielfalt erläutere ich nicht alle Symbole.

Die Symbole der GebärdenSchrift lassen sich entsprechend der Gebärdensprach-Phoneme in Symbole für manuelle und nonmanuelle Komponenten unterteilen. Erstere werden im Folgenden als Handsymbole bezeichnet. In den Handsymbolen und ihren Variationen kommen die in Abschnitt 2.1.1 genannten vier Parameter der manuellen Komponente von Gebärden zum Ausdruck:

- Repräsentation der *Handform*: Alle Handformen der GebärdenSchrift sind von drei Grundformen abgeleitet: die Flachhand, die geschlossene Faust und die offene Faust. Alle weiteren Handformen können durch das Austrecken von Fingern aus diesen abgeleitet werden. Symbolisiert durch entsprechend geformte Linien, bilden diese kombiniert mit den Grundformen nahezu alle weiteren Handsymbole (siehe Abbildung 7).

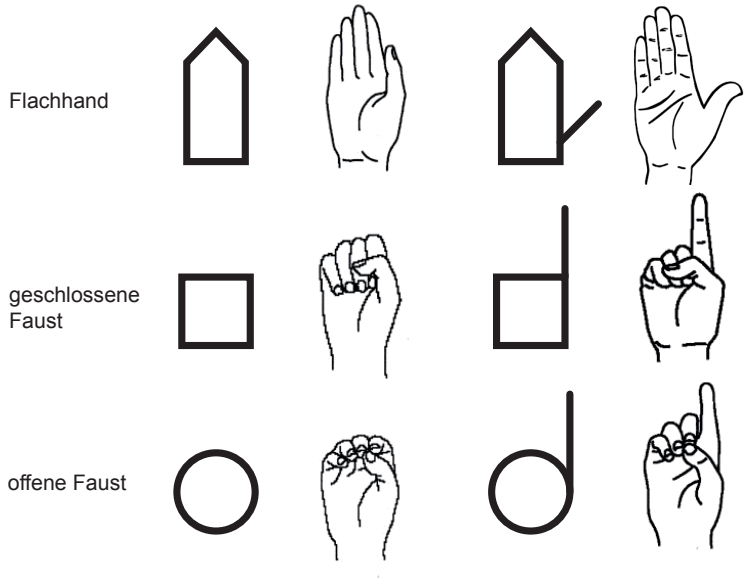


Abbildung 7: Die drei Grundformen der Hand (links) und jeweils eine Variation (rechts). (Quelle: Wöh05)

- Repräsentation der *Handstellung*: Um den dreidimensionalen Raum zweidimensional abbilden zu können, werden in der GebärdenSchrift Ebenen eingeführt. Zwei dieser Ebenen sind besonders entscheidend: Eine horizontale Ebene parallel zum Fußboden oder zu einer Schreibtischoberfläche und eine vertikale Ebene parallel zu einer Wand vor bzw. hinter dem Gebärdenden. Die beiden Ebenen werden im Folgenden aufgrund ihrer Ausrichtung anschaulich als *Schreibtischebene* und als *Wandebene* bezeichnet. Anhand dieser Ebenen lässt sich auch die Ausrichtung der Hand beschreiben. Ist die Hand parallel zur Schreibtischebene ausgerichtet, wird dies in der GebärdenSchrift durch einen Spalt an den Fingergrundgelenken des Handsymbols symbolisiert. In diesem Fall ist die Hand von oben zu betrachten (siehe Abbildung 8 rechts). Ansonsten stellt das Handsymbol eine zur Wandebene parallel ausgerichtete bzw. von hinter dem Gebärdenden betrachtete Hand dar (siehe Abbildung 8 links). Für Ausrichtungen parallel zu diagonalen Ebenen gibt es zusätzliche Variationen der Symbole, die ich hier jedoch nicht vorstelle. Die Drehung innerhalb der Ebenen wird durch Drehung des Symbols auf dem Papier dargestellt (siehe Abbildung 8 unten). Abschließend beschreibt die Füllung des Handsymbols, welche Seite der Hand zum Gebärdenden (Wandebene) bzw. nach oben (Schreibtischebene) zeigt. Bei einem (schwarz) ausgefüllten Handsymbol ist es der Handrücken, bei einem nicht ausgefüllten Handsymbol die Handfläche und bei einem halbausgefüllten Handsymbol die Handkante. In letzterem Fall zeigt die ausgefüllte Hälfte immer zum Handrücken (siehe Abbildung 8).

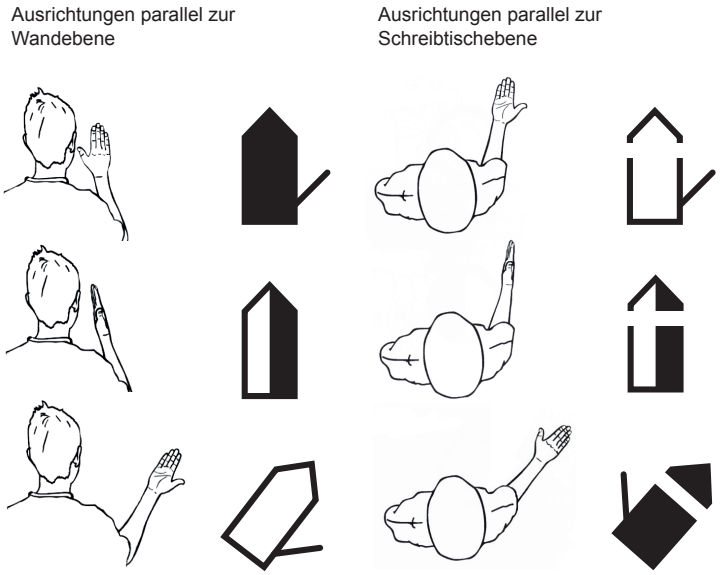


Abbildung 8: Beispiele für verschiedene Handstellungen. (Quelle: vgl. Wöh05, S. 54,55)

- Repräsentation der *Ausführungsstelle*: Die Ausführungsstelle der manuellen Komponente wird anschaulich durch die zweidimensionale Anordnung der Handsymbole beschrieben – in der Regel relativ zum Kopf. Darüber hinaus können die Schulter- und die Hüfthöhe durch (horizontale) Striche beschrieben werden, sofern die Ausführungsstelle für die Darstellung einer Gebärde weiter konkretisiert werden soll. Dies gilt auch für weitere Körperteile wie zum Beispiel für die Arme oder den Hals. Da sehr viele Gebärden an bestimmten Stellen des Kopfes ausgeführt werden, kann die Kontaktstelle am Kopf genauer gekennzeichnet werden, zum Beispiel über das Einzeichnen eines Ohres oder eines Halbkreises, der für Bereiche wie die Kinn- oder Stirnpartie steht.

- Repräsentation der *Bewegung*: Handexterne Bewegungen werden durch Pfeile visualisiert. Diese unterscheiden sich, je nachdem in welcher Ebene die Bewegung ausgeführt wird: Ein Pfeil mit einer doppelten Linie beschreibt eine Bewegung in der Wandebene, ein Pfeil mit einer einfachen Linie eine Bewegung in der Schreibtischebene. Darüber hinaus existieren weitere Pfeile beispielsweise für Bewegungen in weiteren Ebenen oder für Kippbewegungen des Handgelenks. Die Pfeilspitze kennzeichnet jeweils, von welcher Hand die Bewegung ausgeführt wird: Eine (schwarz) ausgefüllte Pfeilspitze steht für die rechte, eine nicht ausgefüllte für die linke Hand. Des Weiteren können Handsymbole durch Symbole ergänzt werden, die handinterne Bewegungen wie zum Beispiel Fingerbewegungen darstellen. Kontaktsymbole beschreiben zusätzlich, ob und in welcher Form bei der Ausführung der Gebärde Berührungen stattfinden. Der Schlagkontakt beschreibt zum Beispiel eine energische, schlagartige Berührung. Weitere Beispiele für Kontakte sind der einfache Kontakt, der Reibekontakt, der Wischkontakt oder der Greifkontakt.

Auch die nonmanuellen Komponenten einer Gebärde können mit Hilfe der Gebärden-Schrift dargestellt werden:

- Repräsentation der *Mimik*: Die GebärdenSchrift bietet Symbole für viele Aspekte der Mimik. So kann zum Beispiel die Stellung der Wangen, der Augenbrauen und der Augen dargestellt werden. Der Kopf wird dabei in der verschriftlichten Gebärde durch einen Kreis dargestellt, die zuvor genannten Aspekte der Mimik durch Symbole an der entsprechenden Stelle innerhalb dieses Kopfsymbols.

- Repräsentation der *Blickrichtung*: Die Blickrichtung wird über Pfeile symbolisiert, die an Stelle der Augen in das Kopfsymbol geschrieben werden. Analog zu den Bewegunspfeilen stehen Pfeile mit doppelter Linie für Blickrichtungen entlang der Wandebene und Pfeile mit einfacher Linie für Blickrichtungen entlang der Schreibtischebene.

- Repräsentation der *Mundbewegungen*: Unter den urpsrünglichen Sutton-SignWriting-Symbolen waren vor allem Symbole zur Beschreibung der Mundgestik. Stefan Wöhrmann hat jedoch erkannt, dass ergänzende Mundbilder das Lesen von Gebärden-Schrift-Texten stark erleichtern. Aus diesem Grund hat er die GebärdenSchrift um entsprechende Symbole ergänzt, welche die zu den Lauten passenden Mundbilder

abbilden. Indem mehrere Kopfsymbole nebeneinander geschrieben werden, lässt sich so auch das komplette Mundbild der Gebärde darstellen.

- Repräsentation der *Kopf- und Oberkörperhaltung*: Zur Beschreibung der Kopf- und Oberkörperhaltung bzw. deren Bewegung bedient sich die GebärdenSchrift der gleichen Prinzipien wie bei der Beschreibung der Handstellung bzw. -bewegung. Um die Schräglage des Kopfes zu beschreiben gibt es zwar ein Symbol, anschaulich kann dies aber auch durch eine Drehung des Kopfsymbols geschehen. Des Weiteren visualisieren beispielsweise Pfeile oberhalb des Kopfsymbols Kopfbewegungen. Ebenso wie bei der Handbewegung oder der Blickrichtung unterscheiden sich die Pfeile, je nachdem in welcher Ebene die Bewegung ausgeführt wird.

Neben den Symbolen zur Beschreibung der Gebärden beinhaltet die GebärdenSchrift auch eigene Satzzeichen. Diese sind allerdings bedeutungsäquivalent zu denen der deutschen Schriftsprache.

2.3.2 Das International SignWriting Alphabet

Das International SignWriting Alphabet (ISWA) ist eine systematische Sammlung aller GebärdenSchrift-Symbole. Analog zum Alphabet der lateinischen Schrift zählt es alle Symbole auf und weist ihnen eine feste Reihenfolge zu. Diese kann beispielsweise zum Sortieren und systematischen Suchen von verschriftlichten Gebärden verwendet werden. Die zahlreichen GebärdenSchrift-Symbole werden innerhalb des ISWA zudem in Kategorien und Gruppen eingeteilt und anhand dessen systematisch benannt. Diese Kategorisierung wird innerhalb des im Rahmen dieser Arbeit entwickelten GebärdenSchrift-Editors verwendet, um die große Menge der Symbole für den Benutzer übersichtlich anzuzeigen. Des Weiteren sind die ISWA-Symbole in einer Onlinedatenbank frei verfügbar[1]. Bei der Anzeige der Symbole innerhalb des GebärdenSchrift-Editors greife ich auf diese Datenbank zu.

Aus diesen Gründen stelle ich das ISWA im Folgenden kurz vor. Das ISWA teilt die GebärdenSchrift-Symbole zunächst in sieben Kategorien ein. Diese Kategorien enthalten unterschiedlich viele Gruppen. Insgesamt gibt es 30 Symbolgruppen. Die Gruppen enthalten wiederum unterschiedlich viele Basissymbole. Deren Anzahl variiert je nach Gruppe zwischen 5 und 58. Insgesamt gibt es 639 Basissymbole. Von diesen Basissymbolen existieren des Weiteren bis zu 96 Variationen. Bei Handsymbolen kommen diese zum Beispiel durch die Kombination der in Abschnitt 2.3.1 beschriebenen Füllungen und Rotationen zu Stande. Insgesamt ergibt sich so eine Gesamtzahl von 35.023 Symbolen. (Vgl. Sle09) Das ISWA enthält dabei jedoch die Symbole zur Beschreibung verschiedenster Gebärdensprachen auf der ganzen Welt. Zur Verschriftlichung der DGS ist eine kleinere Menge von Symbolen ausreichend.

Die Systematisierung der Symbole ist vergleichbar mit einer Baumstruktur. Die Blätter repräsentieren dabei die Symbole, die restlichen Knoten die Kategorien, Gruppen und Basissymbole. Die Einteilung und Sortierung erfolgt nach bestimmten Prinzipien.

[1] `http://www.signbank.org/swis/data.php?subset=\&sg_code=*`

Die erste der sieben Kategorien enthält alle Handsymbole, also die Symbole zur Beschreibung des Handformparameters. Diese sind des Weiteren in zehn Gruppen aufgeteilt. Die Aufteilung orientiert sich dabei an den Gebärden der American Sign Language (ASL) für die Zahlen von Eins bis Zehn. In der ersten Gruppe sind zum Beispiel alle Handformen mit einem ausgestreckten Zeigefinger, da die Zahl „Eins“ in ASL durch eine Faust mit ausgestrecktem Zeigefinger repräsentiert wird.

Die zweite Kategorie enthält alle Symbole zur Beschreibung des Bewegungsparameters, die ebenfalls in zehn Gruppen aufgeteilt sind. In die ersten beiden Gruppen fallen Kontaktsymbole und Symbole zur Beschreibung von Fingerbewegungen. Die folgenden Gruppen beinhalten Bewegungspfeile. Deren Aufteilung richtet sich zum einen nach der Ebene, in der die repräsentierte Bewegung ausgeführt wird. Zum anderen wird zwischen geraden und gekrümmten Bewegungen unterschieden.

Die weiteren Kategorien beinhalten Gesicht- und Kopfsymbole, Symbole zu Beschreibung des Körpers, Dynamiksymbole (Eigenschaften von Bewegungen), Satzzeichen und Symbole zur genaueren Verortung im Raum. Deren Aufteilung in Gruppen erläutere ich an dieser Stelle nicht näher.

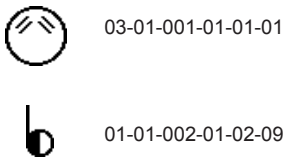


Abbildung 9: Zwei GebärdenSchrift-Symbole und ihr Identifikationscode im ISWA.

Entsprechend ihrer Systematisierung wird jedem Symbol wie zuvor bereits angedeutet ein Identifikationscode zugeordnet (vgl. Sle09). Dieser besteht aus sechs mit Bindestrichen verbundenen Nummern, wobei die dritte dreistellig und die restlichen zweistellig sind. Abbildung 9 zeigt zwei Beispiele. Die erste der Nummern steht für die Kategorie, die zweite für die Gruppe, wobei die Gruppennummerierung in jeder Kategorie erneut bei eins beginnt. Die dritte dreistellige Nummer ist eine fortlaufenden Nummerierung der Symbole innerhalb einer Gruppe. Die vierte Nummer beschreibt Variationen, also leichte Abwandlungen ansonsten ähnlicher Smybole – zum Beispiel verschiedene Neigungen des gleichen Fingers. Die letzten zwei Nummern kodieren verschiedene Ausprägungen des gleichen Symbols und werden in jeder Gruppe anders verwendet. Bei Handsymbolen beschreiben sie die in Abschnitt 2.3.1 beschriebenen Repräsentationen der Handstellung. In diesem Fall steht die vorletzte Nummer für die Füllung (beinhaltet die Orientierung in der Wand- und Schreibtischebene) und die letzte Nummer für die Rotation (beinhaltet auch Spiegelungen des Symbols). Dies ist bei Bewegungspfeilen ähnlich: Sie können ebenfalls rotiert und ihre Pfeilspitzen unterschiedlich gefüllt werden. Bei den Kopfsymbolen werden die letzten beiden Nummern verwendet, um zum Beispiel Symbole mit zwei eingezeichneten Augen, mit einem eingezeichneten Auge oder ohne Umrandung des Kopfes zu unterscheiden. Diese Benennung der Symbole habe ich bei der Entwicklung des GebärdenSchrift-Editors übernommen.

2.3.3 Andere Notationsformen für Gebärden

Dieser Abschnitt dient der Einordnung der GebärdenSchrift vor dem Hintergrund anderer Notationsformen für Gebärden. Die Gemeinschaft derer, die sich mit Gebärdensprachen beschäftigen, ist sich keineswegs einig, welche die ideale Notationsform für Gebärden ist. Diese Frage kann und möchte ich in diesem Abschnitt nicht beantworten. Ich stelle lediglich einige Notationsformen wertfrei vor:

Zeichnungen In dieser Arbeit verwende ich neben der GebärdenSchrift Zeichnungen, um Gebärden zu veranschaulichen. Zu ihrem Verständnis ist kaum Vorwissen notwendig. Mit Hilfe verschiedener Pfeilformen können anschaulich Bewegungen dargestellt werden. Die Anfertigung dementsprechender Abbildungen setzt jedoch gute Zeichenfertigkeiten voraus und nimmt viel Zeit in Anspruch. Bei komplexen Bewegungen wird es immer schwieriger, mit nicht-formalen Mitteln eine verständliche und eindeutige Abbildung zu erzeugen.

Glossentransskription Eine Glosse ist eine formale Repräsentation der Gebärde durch ein oder mehrere lautsprachliche Wörter. In den meisten Fällen entspricht die Glosse zu einer Gebärde auch ihrer Bedeutung. Die Glosse wird in Großdruckbuchstaben geschrieben, um sie von einem Wort der jeweiligen Schriftsprache abzugrenzen. So bezeichnet „VATER“ die Gebärde für „Vater“. Durch zusätzliche Parameter werden Eigenschaften der Gebärde beschrieben. „P-ort-GEBEN-P-ort“ steht zum Beispiel für die Gebärde für „geben“ (siehe Abbildung 10). Da diese Gebärde zu den in Abschnitt 2.1.2 erläuterten Transferverben gehört, beginnt und endet die Gebärde räumlich bei einer Person, was durch die Parameter „P-ort“ beschrieben wird. Die Glossentransskription benötigt zur Darstellung keinen speziellen Zeichensatz, lässt sich also wie in diesem Abschnitt in lautsprachlichen Text einbinden. Die Glosse beschreibt jedoch kaum die Gestalt der Gebärde, sie ist vielmehr eine formale Beschreibung der Gebärde aus lautsprachlicher Sicht. Mit ihrer Hilfe lässt sich zum Beispiel die Satzstellung der DGS beschreiben, jedoch nur für Personen, welche die jeweilige Lautsprache bzw. die zugehörige Schriftsprache beherrschen. (Vgl. PvMMH08, S. 3–5)

Im in dieser Arbeit beschriebenen Übersetzungskontext wird einer verschriftlichten Gebärde ebenfalls eine lautsprachliche Bedeutung zugeordnet. Diese wird als Schlüssel in der Datenbank verwendet, um die Gebärde beim Übersetzen anhand des zu übersetzenden Wortes zu suchen. Die Glossentransskription kann dabei genutzt werden, um die lautsprachliche Bedeutung formaler zu beschreiben, wie zum Beispiel den Unterschied zwischen verschiedenen Flexionen eines Transferverbs.

HamNoSys Am Institut für deutsche Gebärdensprache der Universität Hamburg wurde das Hamburger Notationssystem (HamNoSys) entwickelt. HamNoSys baute in der ersten Version auf einem Notationssystem nach dem US-amerikanischen Linguisten William Sokoe auf. Mit HamNoSys lässt sich überwiegend die manuelle Komponente einer Gebärde beschreiben. Ebenso wie in der GebärdenSchrift existieren Symbole für die in Abschnitt

Abbildung 10: Die Gebärde für „geben“ in der Variante „P-ich-GEBEN-P-du“. (Quelle: BKPZ07, S. 69)

2.1.1 beschriebenen einzelnen Parameter der manuellen Komponente. Diese werden jedoch im Gegensatz zur GebärdenSchrift sequentiell aufgeschrieben. Abbildung 11 zeigt ein Beispiel. Die sequentielle Beschreibung sorgt bei komplizierten Gebärden für lange, weniger ikonische Zeichenketten, die sich maschinell jedoch sehr gut verarbeiten lassen. (Vgl. fDGuKG04)

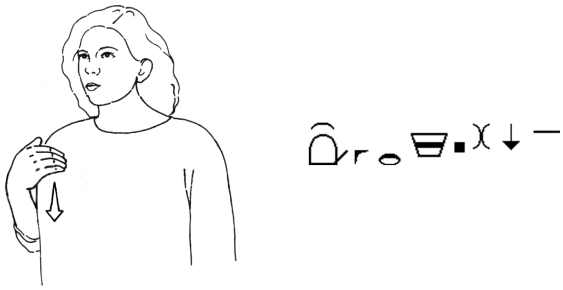


Abbildung 11: Die Gebärde für „mühsam“ notiert in HamNoSys (rechts). (Quelle Zeichnung: BKPZ06, S. 69)(Quelle HamNoSys-Schriftzug: fDGuKG04)

Im folgenden Abschnitt beschreibe ich die in der Einleitung dieser Arbeit angedeutete Problemstellung genauer und gehe dabei auch auf die Rolle der in den vorherigen Abschnitten vorgestellten GebärdenSchrift ein.

2.4 Probleme an Gehörlosenschulen/Anwendungskontext

Dieser Abschnitt vertieft die in der Einleitung vorgestellte Problemstellung. Er beschreibt den Anwendungskontext für den entwickelten GebärdenSchrift-Editor. Dazu gebe ich einen kurzen Überblick über die Geschichte der Hörgeschädigtenpädagogik.

Die Gebärdensprache existiert in Deutschland seit es Gehörlose gibt. Dennoch ist sie in Deutschland erst seit 2002 rechtlich als Sprache anerkannt. Mit dem Behindertengleichstellungsgesetz wurde damals zum Beispiel die Grundlage für ein Anrecht Gehörloser auf einen Dolmetscher geschaffen (vgl. fJ02, §6).

Die im Vergleich zu anderen Ländern späte Anerkennung der Gebärdensprache ist das Resultat eines lang andauernden Methodenstreits über die Schulbildung Gehörloser in der Bundesrepublik Deutschland. In diesem teils noch andauernden Streit geht es um zwei konkurrierende Ansätze der Hörgeschädigtenpädagogik: Die *orale Methode* und den *bilingualen Unterricht*.

- Das zentrale Ziel der *oralen Methode* ist die Eingliederung gehörloser und schwerhöriger Kinder in die hörende, sprechende Gesellschaft. Konkret bedeutet dies, dass gehörlose und schwerhörige Kinder „die Lautsprache unter Berücksichtigung der minimalsten Hörreste“ durch das Ablesen von Lippenbewegungen und den Sprechaufbau über ständige Artikulationsübungen erlernen sollen. Dabei wird die Gebärdensprache „als Hindernis für die Lautsprachanbahnung“ angesehen und wurde deshalb an Schulen teilweise verboten. Ihr Einsatz führe „zu deren Dominanz und gleichermaßen zur Verkümmerung der Lautsprache“. Kritiker dieses Ansatzes bezeichnen ihn als defizitorientiert. Gehörlose werden innerhalb der Welt der Hörenden ihr Leben lang „kommunikationsbehindert“ bleiben. (Alle Zitate vgl. BS07)

- Der *bilinguale Unterricht* bezieht im Gegensatz dazu die Gebärdensprache mit ein. Ziel ist die Erziehung sowohl in der DGS als auch in der deutschen Lautsprache bzw. speziell in der Schriftsprache. Die deutsche Schriftsprache soll in diesem Ansatz wie eine Fremdsprache auf Basis einer soliden Muttersprache, nämlich der Gebärdensprache, erlernt werden. Die Gebärdensprache hat dabei zudem eine „besondere Bedeutung für die psychische, soziale und kulturelle Identitätsfindung der Gehörlosen in der Gehörlosengemeinschaft“ (vgl. BS07).

Gehörlose sind oft Kinder hörender Eltern und wachsen daher nicht zwangsläufig mit der Gebärdensprache auf. Mit Hilfe der fortschreitenden Technik in Form von Hörgeräten oder Cochleaimplantaten sowie der frühen Diagnostik werden heute bereits „beträchtliche Ergebnisse im Bereich der Sprach- und Sprechkompetenz erzielt“ (vgl. BS07). Cochleaimplantate sind Hörprotesen für Gehörlose, deren Hörnerv noch funktioniert. Der in Abschnitt 1.1 vorgestellte Stefan Wöhrmann unterrichtet allerdings Gehörlose und schwerhörige Schüler, die trotz der verbesserten technischen Möglichkeiten die entscheidende Phase des Spracherwerbs verpasst haben, da ihre Eltern nicht oder nur kaum mit ihnen kommunizieren konnten. Im bilingualen Unterricht wird diesen Kindern mit der Gebärdensprache ermöglicht, überhaupt erst einmal die Kommunikation mit anderen zu erlernen und dies in einer Sprache, in der sie nicht von vornherein aufgrund ihres mangelnden oder eingeschränkten Gehörs benachteiligt sind.

Der im Rahmen dieser Arbeit entwickelte GebärdenSchrift-Editor soll Lehrer im bilingualen Unterricht unterstützen. Dies gilt primär für den Unterricht in der deutschen Schriftsprache. Wie zuvor und in der Einleitung dieser Arbeit bereits angedeutet gleicht dieser Lernprozess dem Erlernen einer Fremdsprache mit unterschiedlichen Schriftsymbolen.

Die DGS ist dabei die Ausgangssprache, die deutsche Laut-/Schriftsprache die Zielsprache. Die deutsche Schriftsprache ist eine Verschriftlichung der deutschen Lautsprache. Die Symbole der lateinischen Schrift haben einen Bezug zu den Sprechlauten. Diese nehmen Gehörlose und Schwerhörige jedoch nicht (vollständig) wahr. Die Buchstaben der deutschen Schriftsprache haben für Gehörlose daher keine tiefere Bedeutung. Sie müssen die Schreibweise von Wörtern auswendig lernen, ohne ihren Klang als zusätzlichen Bezug zur Herleitung der Schreibweise zu kennen.

Der Vergleich mit dem Erlernen einer Fremdsprache ist darüber hinaus insofern passend, als es dabei vor allem nötig ist, eine neue Grammatik zu erlernen. In Abschnitt 2.1 habe ich bereits die entscheidenden Unterschiede zwischen der DGS und der deutschen Lautsprache erläutert. Sie verdeutlichen, dass es sich bei der DGS nicht um eine andere Repräsentation der deutschen Lautsprache, sondern um eine eigenständige Sprache mit einer anderen Grammatik handelt. Die grammatikalischen Unterschiede zwischen der DGS und der deutschen Lautsprache fallen dabei größer aus als zum Beispiel zwischen verschiedenen romanischen Sprachen. Die zunächst fremde Grammatik der deutschen Lautsprache muss von den gehörlosen Schülern erst erlernt werden. Dieser Prozess ist aufgrund der großen Unterschiede beider Sprachen sehr mühsam. Wie in Abschnitt 2.2 erläutert, kann die Grammatik der deutschen Lautsprache mit den Mitteln der Gebärdensprache dargestellt werden, um die Schüler bei der Verknüpfung beider Sprachen zu unterstützen. Wie beschrieben dienen diese lautbegleitenden Gebärden als Mittlerfunktion zwischen den beiden Sprachen.

Der GebärdenSchrift kommt dabei eine wichtige Rolle zu. Um die gehörlosen Schüler beim Erlernen und Kontrastieren zweier Sprachen zu unterstützen, müssen diese und die zwischen ihnen vermittelnde LBG vergegenständlicht werden. Im herkömmlichen Fremdsprachunterricht ist es selbstverständlich, dass Texte, Vokabellisten und Grammatikbeispiele den Schülern in der Ausgangs- und Zielsprache gegenübergestellt präsentiert werden. Die GebärdenSchrift bietet nun die Möglichkeit, dies auch im Zusammenhang mit der DGS zu tun. Das einfache Vorführen im Unterricht ermöglicht den Schülern keine Konservierung der Ergebnisse zu deren Nachbereitung. Mit Hilfe der GebärdenSchrift können jedoch Vokabellisten, Grammatikübungen, Diktate usw. erstellt werden. Abbildung 12 zeigt ein Diktat in GebärdenSchrift. Der dargestellte Text ist ein Text in lautsprachbegleitender Gebärdensprache. Aufgabe des Schülers war es, den Text in deutscher Schriftsprache zu ergänzen. Bei der Erstellung kann ein lautsprachlicher Text wie zum Beispiel ein Märchen als Vorlage dienen.

Im Gegensatz zu beispielsweise vorgefertigten Zeichnungen oder Videos von Gebärden ist Stefan Wöhrmann mit Hilfe der GebärdenSchrift in der Lage, auf die individuellen Schwächen seiner Schüler einzugehen. Dabei kann es vorkommen, dass er während des Unterrichts für bestimmte Schüler zusätzlich Unterrichtsmaterial erstellt oder bestehende Texte abwandelt, um individuelle Hilfestellungen zum Beispiel in Form von ausführlichen Mundbildern zu ergänzen.

Ebenso wie beim Erlernen der Schriftsprache kann die GebärdenSchrift auch den umgekehrten Lernprozess unterstützen: Das Erlernen der DGS auf Basis der deutschen Lautsprache. Aus eigener Erfahrung weiß ich, dass mir im DGS-Unterricht die Möglichkeit fehlt, eingeführte Gebärden zu notieren, um diese später wiederholen zu können. Das

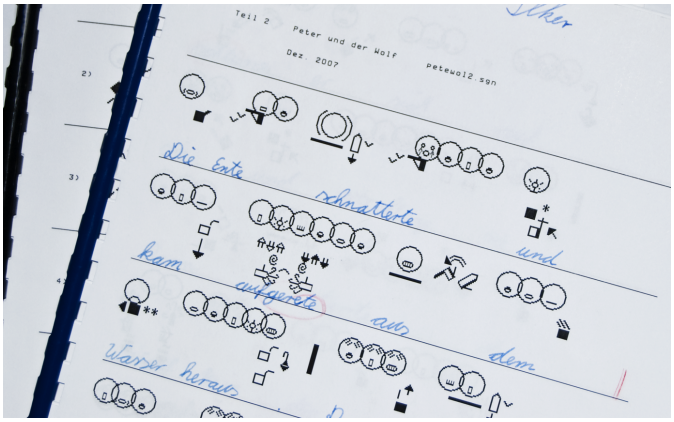


Abbildung 12: Ein Diktat in GebärdenSchrift. (Quelle: Stefan Wöhrmann)

Lehrbuch verwendet zwar anschauliche Zeichnungen zur Darstellung der Gebärden, dieser Wortschatz ist jedoch begrenzt. Denn im Laufe einer individuellen Unterrichtsstunde werden viele DGS-Gebärden eingeführt, die nicht im Lehrbuch abgebildet sind. Die GebärdenSchrift bietet einem Kursleiter die Möglichkeit, schnell individuelle Vokabellisten zu erstellen, ebenso Übungsblätter und Grammatikbeispiele.

3 Anforderungsanalyse

In Abschnitt 2.4 habe ich beschrieben, inwiefern die GebärdenSchrift im Unterricht an Gehörlosenschulen zum Einsatz kommt. Die Erstellung der dort aufgeführten Unterrichtsmaterialien soll nun durch Software unterstützt werden. Das zentrale Problem bei der Erstellung von Unterrichsmaterial in GebärdenSchrift entsteht durch die zu Beginn von Abschnitt 2.3 beschriebene Besonderheit der GebärdenSchrift. Wie dort schon erwähnt verhindert die zweidimensionale Anordnung ihrer Symbole, dass GebärdenSchrift-Texte mit gängigen Textverarbeitungsprogrammen getippt werden können und für ihre Erstellung ein spezielles Programm notwendig ist. Dieses Kapitel beschreibt die Anforderungen an einen solchen GebärdenSchrift-Editor und untersucht, inwiefern aktuelle, entsprechende Programme diese Anforderungen erfüllen.

3.1 Anforderungen

In diesem Abschnitt leite ich aus dem in Abschnitt 2.4 beschriebenen Anwendungskontext Anforderungen für einen GebärdenSchrift-Editor ab. Diese werden später zur Bewertung aktueller GebärdenSchrift-Software verwendet und dienen als Grundlage für den in Kapitel 4 beschriebenen Entwurf eines solchen Editors.

Bevor ich mit der Definition von Anforderungen beginne, möchte ich kurz beschreiben, wer als Zielgruppe für den GebärdenSchrift-Editor in Frage kommt. Ohne diese festzusetzen, ist es nicht sinnvoll, Anforderungen aufzustellen. Denn diese hängen maßgeblich davon ab, mit welchen Vorkenntnissen ein Benutzer den GebärdenSchrift-Editor bedient. Zuvor habe ich beschrieben, dass die Software hauptsächlich an Gehörlosenschulen bei der Erstellung von Unterrichtsmaterial zum Einsatz kommen soll. Damit steht die primäre Benutzergruppe fest: Lehrende an Gehörlosenschulen. Ich gehe des Weiteren davon aus, dass Benutzer der Software über Kenntnisse sowohl der DGS als auch der deutschen Lautsprache verfügen. Das Programm dient als Hilfestellung bei der Übersetzung in GebärdenSchrift, automatisiert diese jedoch nicht vollständig.

Aus verschiedenen Gründen ist es sinnvoll, dass auch Gehörlose beim Erlernen der deutschen Schriftsprache mit dem GebärdenSchrift-Editor arbeiten. Die Bedienung soll jedoch zunächst aus Sicht der Lautsprache erfolgen. Dies betrifft zum Beispiel die Bennennung von Menüs oder die Suche in der Gebärdendatenbank. Gehörlose ohne Kenntnisse der deutschen Schriftsprache sind daher zunächst nicht primär als Benutzergruppe vorgesehen. Denn dies würde gerade im Fall von gehörlosen Schülern ein ganz anderes Bedienkonzept erfordern. Dies schließt jedoch nicht aus, dass Schüler den GebärdenSchrift-Editor nutzen können, sofern sie über fortgeschrittene Schriftsprachkenntnisse verfügen oder durch einen Lehrenden angeleitet werden. Des Weiteren könnten sie die im Folgenden beschriebene Übersetzungsfunktion zum Nachschlagen von laut- bzw. schriftsprachlichen Vokabeln nutzen.

Die folgenden Abschnitte beschreiben die einzelnen Anforderungen. Dabei begründe ich jeweils, warum ich gerade diese Anforderungen aufgestellt habe.

Abspeichern und Einfügen vordefinierter verschriftlichter Gebärden Aufgrund der zweidimensionalen Anordnung der GebärdenSchrift-Symbole ist es aufwändiger, sie zu Gebärden zu kombinieren. Während die Buchstaben des lateinischen Alphabets einfach linear hintereinander geschrieben werden können, geht bei der Platzierung der einzelnen GebärdenSchrift-Symbole viel Zeit verloren. Die Erstellung von GebärdenSchrift-Texten dauert also sehr lange, wenn nur einzelne Symbole zur Verfügung stehen und ist daher meistens nicht sinnvoll. In der Lautsprache tritt fast jedes Wort aufgrund von Konjugation und Deklination im Satzzusammenhang in flektierter Form auf. Zwar habe ich in Abschnitt 2.1.2 einige Mechanismen vorgestellt, die zu einer Anpassung von DGS-Gebärden im Satzzusammenhang führen, diese sind jedoch seltener als in der deutschen Lautsprache. Dies führt dazu, dass komplette Gebärden häufig wieder verwendet werden können, um Texte zu schreiben. Das heißt, es ist viel effizienter GebärdenSchrift-Texte zu bilden, indem fertige Gebärden aus einem Wörterbuch ausgewählt und aneinandergereiht werden.

Aus diesen Gründen soll ein GebärdenSchrift-Editor die Möglichkeit bieten, schnell vordefinierte Gebärden zu erstellen, abzuspeichern und einzufügen.

Unterstützung bei der Übersetzung in LBG/DGS In dem in Abschnitt 2.4 beschriebenen Anwendungskontext dient die GebärdenSchrift vor allem dazu, die DGS mit der Lautsprache zu verknüpfen. Wie zuvor beschrieben gehe ich davon aus, dass die Software

hauptsächlich von hörenden Lehrenden genutzt wird, die GebärdenSchrift-Texte vor allem anhand der lautsprachlichen Bedeutung erstellen. Das heißt, sie übersetzen einen lautsprachlichen Text in LBG oder in DGS – repräsentiert in GebärdenSchrift. Dabei suchen sie Gebärden in der Datenbank vor allem anhand ihrer lautsprachlichen Bedeutung. Eine effektive Unterstützung dieses Übersetzungsvorgangs ermöglicht die Nutzung der unzähligen, existierenden, lautsprachlichen Unterrichtsmaterialien und Texte – von klassischen Märchen hin zu aktuellen Internet-Artikeln.

In Abschnitt 2.2 habe ich beschrieben, dass die Übereinstimmung der Satzstellung zwischen der LBG und der deutschen Lautsprache eine automatische Übersetzung begünstigt, diese sich jedoch aufgrund von Synonymen nicht ohne Weiteres vollständig automatisieren lässt. Ein bei der Übersetzung unterstützender GebärdenSchrift-Editor sollte daher den Prozess der Übersetzung nur so weit wie möglich automatisieren. Das heißt, es kann zu den Wörtern eines eingegebenen lautsprachlichen Satzes automatisch nach Übersetzungen gesucht werden. Dem Benutzer soll dann für jedes Wort eine Auswahl möglicher Übersetzungen präsentiert werden, unter denen er anschließend die zum Kontext passende auswählt.

Der lautsprachliche Text soll sich mit den gewohnten Editorfunktionen bearbeiten lassen. Das bedeutet, es ist möglich, Text zu markieren, zu kopieren und an anderer Stelle einzufügen. Darüber hinaus soll auf diese Weise auch Text aus anderen Quellen eingefügt werden können, um bestehende lautsprachliche Texte in DGS oder LBG repräsentiert in der GebärdenSchrift zu übersetzen. Alle Änderungen des lautsprachlichen Textes sollen automatisch zur Anpassung des entsprechenden GebärdenSchrift-Textes führen. Werden dabei Wörter nicht geändert, soll deren Repräsentation in GebärdenSchrift ebenfalls unverändert bleiben. Dies gilt insbesondere, wenn der Benutzer dort bereits eine Übersetzungsalternative ausgewählt hat.

Erstellen und Editieren verschriftlichter Gebärden Auch wenn vordefinierte Gebärden das Schreiben von GebärdenSchrift-Texten erleichtern, ist es aus verschiedenen Gründen notwendig, die einzelnen Symbole, aus denen sie bestehen, zu ändern bzw. Gebärden neu zu erstellen.

Zum einen entstehen regelmäßig neue Gebärden, ebenso wie in der Lautsprache neue Wörter gebildet werden. Dies geschieht zum Beispiel bei Eigennamen, die in der Öffentlichkeit plötzlich eine wichtige Rolle spielen – so zum Beispiel beim Amtsantritt des US-Präsidenten „Barac Obama“. Die Online-Datenbank von SignPuddle bietet zwar eine umfangreiche Datenbank mit Gebärden, diese enthält jedoch nur ca. 11.000 Einträge (siehe Abschnitt 3.2.3). Darüber hinaus ist es nahezu unmöglich, wirklich alle Gebärden der DGS zu erfassen. Der zu erstellende GebärdenSchrift-Editor benötigt daher zwangsläufig eine Möglichkeit, neue Gebärden aus einzelnen GebärdenSchrift-Symbolen zu erstellen, um die Gebärdendatenbank zu erweitern.

Zum anderen habe ich in Abschnitt 2.1.2 verschiedene Prinzipien der Gebärdenbildung und ihrer Flexion erläutert. Sie beschreiben, wie Gebärden in der DGS im Satzkontext gebildet oder verändert werden. Ein Beispiel waren die Transferverben, die je nach Satzzusammenhang abgewandelt werden. Die Anpassung ist dabei systematisch, eine Abspeiche-

rung aller möglichen Ausprägungen in der Datenbank auch möglich aber nicht unbedingt hilfreich. Ebenso können neue Gebärden durch die Abwandlung bereits existierender Gebärden gebildet werden. Dies ist zum Beispiel besonders hilfreich, wenn Gebärden durch die ebenfalls in Abschnitt 2.1.2 beschriebene Inkorporation entstehen. Denn in diesem Fall werden Elemente aus vorhandenen Gebärden kombiniert. Darüber hinaus kann die Änderung einer vorhandenen Gebärde aus anderen Gründen erfolgen. Wie in Abschnitt 2.4 beschrieben, ist es zudem sinnvoll, spezielle Änderungen an Gebärden vorzunehmen, um Schülern eine Hilfestellung zu geben. Deshalb müssen Gebärden in der Datenbank strukturerhaltend gespeichert werden. Einzelne Symbole sollen sich bei der späteren Verwendung manipulieren oder austauschen lassen.

Beim Erstellen und Editieren sollen Komfortfunktionen die Arbeit erleichtern: Symbole können per Copy & Paste wiederverwendet werden. Es soll auch möglich sein, mehrere Symbole zu selektieren und gemeinsam zu verschieben oder zu kopieren.

Editieren der Gebärden während der Texterstellung Sowohl das Erstellen als auch das Abändern von Gebärden soll direkt während der Texterstellung geschehen können. Müsste dafür in einen gesonderten Dialog oder gar ein extra Porgramm gewechselt werden, unterbräche dies den Arbeitsfluss unnötigerweise.

Komfortabler Zugriff auf die GebärdenSchrift-Symbole In Abschnitt 2.3.2 habe ich beschrieben, dass das International SignWriting Alphabet (ISWA) über 35000 Symbole enthält. Zum Erstellen von DGS-Texten ist jedoch nur ein Teil davon notwendig. Der Symbolsatz sollte also entsprechend regional angepasst werden. Dennoch bleibt eine große Anzahl von Symbolen übrig. Diese sollten sinnvoll gruppiert werden, sodass ein Benutzer ein gesuchtes Symbol schnell findet. Die Gruppen und Kategorien des ISWA bilden dabei einen Standard. Dennoch sollte bei der Gruppierung beachtet werden, dass ein Benutzer nicht zu viele Gruppen durchlaufen muss, um zum gesuchten Symbol zu gelangen. Dies würde nämlich den Aufwand zum Einfügen der Symbole erhöhen. Die Navigation innerhalb der Symbolgruppen und -kategorien soll anschaulich visualisiert werden, sodass der Benutzer stets weiß, in welcher Gruppe oder Kategorie er sich befindet und welche es noch gibt.

Ein Großteil der Symbole des ISWA sind zudem Variationen der Grundsymbole. Die in Abschnitt 2.3.2 erwähnte Datenbank führt sie als jeweils einzelne Symbole (extrinsische Datenbank). Dies erhöht ihre Anzahl erheblich. Einfacher zu handhaben, wäre eine intrinsische Datenbank, in der nur die Grundsymbole gespeichert sind und deren Variationen durch Operationen gebildet werden können. Letzteres kann dem Benutzer als eine Sicht auf die extrinsische Datenbank präsentiert werden. Während der Benutzer ein Symbol beispielsweise dreht, wird im Hintergrund das Symbol mit der entsprechend nächsten gültigen Drehung ausgewählt und angezeigt.

Anschauliche und selbsterklärende Steuerung Die Steuerung des GebärdenSchrift-Editors ist das zentrale Qualitätsmerkmal. Es ist wichtig, dass sie anschaulich und selbsterklärend gestaltet wird. Auf diese Weise soll Benutzern das schnelle Erlernen der Steuerung

ermöglicht werden. Denn es ist ein zentrales Anliegen dieser Anwendung, die Hemmschwelle für den Einsatz der GebärdenSchrift im Unterricht zu senken. Eine komplizierte Steuerung würde den Einsatz jedoch verkomplizieren und hätte einen gegenteiligen Effekt. Unter den zukünftigen Benutzern sind vor allem Personen, die nicht zwangsläufig große Erfahrung im Umgang mit Computern besitzen. Dennoch kann sich nahezu jeder etwas unter einem einfachen Text-Editor vorstellen. Nahezu jeder weiß, wie mit Hilfe der Tastatur lateinische Buchstaben getippt werden. Eine ähnlich simple Steuerung soll auch das elektronische Schreiben in GebärdenSchrift erleichtern.

Effiziente Steuerung Neben der Anschaulichkeit muss die Steuerung vor allem effizient sein. Effizienz ist in diesem Fall auf die Dauer der Erstellung von GebärdenSchrift-Texten bezogen: Der Einsatz des GebärdenSchrift-Editors soll diese beschleunigen, damit wie in Abschnitt 2.4 beschrieben auch während des Unterrichts Materialien in GebärdenSchrift produziert werden können. Dabei sind folgende Kriterien entscheidend: Wie viele Tastendrücke/Mausklicks sind notwendig, um die zuvor beschriebenen Anforderungen umzusetzen, also zum Beispiel das Einfügen, Erstellen oder Ändern vordefinierter Gebärden?

Zur anschließenden Referenzierung fasse ich die Anforderungen in einem nummerierten Kriterienkatalog zusammen:

1. Abspeichern und Einfügen vordefinierter verschriftlichter Gebärden
2. Unterstützung bei der Übersetzung in LBG/DGS
3. Erstellen und Editieren verschriftlichter Gebärden
4. Editieren der Gebärden während der Texterstellung
5. Komfortabler Zugriff auf die GebärdenSchrift-Symbole
6. Anschauliche und selbsterklärende Steuerung
7. Effiziente Steuerung

Diese Kriterien dienen im Folgenden für die Bewertung aktueller Software im Zusammenhang mit der GebärdenSchrift. Diese erfolgt im nächsten Abschnitt.

3.2 Aktuelle GebärdenSchrift-Software

Im vorherigen Abschnitt habe ich meine Anforderungen an einen GebärdenSchrift-Editor definiert. Es existieren bereits Anwendungen, die das Erstellen von GebärdenSchrift-Texten ermöglichen. Diese werden im Folgenden vorgestellt. Ich untersuche dabei, inwieweit sie die definierten Anforderungen erfüllen. Darüber hinaus dienen einige Aspekte dieser Programme als Vorbild für den in Abschnitt 4 vorgestellten Entwurf.

3.2.1 SignWriter DOS

SignWriter DOS[2] war der erste Texteditor für GebärdenSchrift. Dem Namen entsprechend läuft die Anwendung nur unter dem veralteten Betriebssystem DOS. Mit Hilfe eines entsprechenden Emulators wie der DOSBox kann SignWriter DOS zwar auch unter neueren Betriebssystemen verwendet werden. Allerdings unterliegt es dann ebenso den DOS-inhärenten Einschränkungen wie Modalität von Dialogen, fehlender Mausunterstützung und monochromer Grafik in niedriger Auflösung (siehe Abbildung 13 und 14). Trotz der stark veralteten Technik wird SignWriter DOS immer noch eingesetzt.

Abbildung 13: Die leere Schreibfläche von SignWriter DOS.

Legt man mit SignWriter DOS ein neues Dokument an, erscheint die in Abbildung 13 gezeigte Schreibfläche. Nun besteht die Möglichkeit einzelne GebärdenSchrift-Symbole über die Tastatur zu „tippen“. Die zahlreichen Symbole der GebärdenSchrift sind in Gruppen geordnet, die sich nur zum Teil an den Symbolgruppen des in Abschnitt 2.3.2 beschriebenen International SignWriting Alphabet orientieren. Um ein Symbol zu tippen, sind daher drei Tastendrücke notwendig. Mit dem ersten wird die Symbolgruppe geöffnet. Der zweite wählt das Symbol innerhalb der Gruppe aus, der dritte die spezifische Ausprägung des zuvor gewählten Symbols, also zum Beispiel verschiedene Drehungen eines Handsymbols. Diese letzte Ebene ist eigentlich nicht notwendig, da das Symbol auch anschließend über eine Taste zum Beispiel in der Rotation angepasst werden kann. Anforderung 5 ist deshalb nur teilweise erfüllt. Die Auswahl des Symbols wird kaum visualisiert. Die Zuordnung der Tasten zu den Symbolgruppen wird nur in der beigefügten Anleitung erläutert. Nach dem Öffnen einer Gruppe visualisiert eine Leiste am unteren Bildschirmrand die Zuordnung der Symbole zu den Tasten der mittleren Buchstabenreihe auf der Tastatur (siehe Abbildung 14). Um welche Tasten es sich handelt, steht ebenfalls nur in der Anleitung. Ein großer Vorteil: Für verschiedene Länder existieren individuelle

[2] `http://www.signwriting.org/forums/software/sw44/`

Tastaturbelegungen. Dies verbessert die Bewertung im Hinblick auf Anforderung 5, da die Menge der Symbole auf die relevanten eingeschränkt wird.

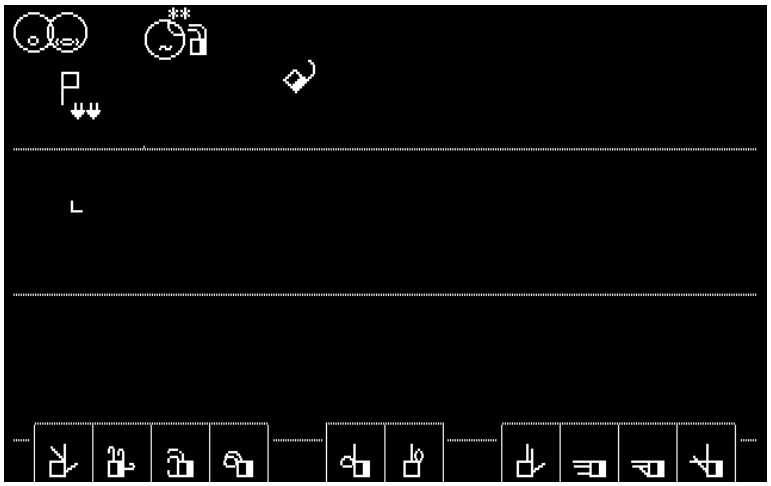


Abbildung 14: Tippen in GebärdenSchrift mit SignWriter DOS.

Die einzelnen Symbole einer Gebärde können selektiert werden. Dazu muss jedoch erst der Cursor mit den Pfeiltasten zum Symbol bewegt werden, was sehr zeitaufwändig ist. Ein eingefügtes Symbol ist immer selektiert. Selektierte Symbole lassen sich mit Hilfe der Pfeiltasten positionieren. Des Weiteren gibt es Tasten, die selektierte Symbole anpassen, zum Beispiel in Größe, Füllung und Rotation. Es ist zudem möglich, Symbole zu kopieren und mehrere Symbole gleichzeitig zu verschieben. Dabei können jedoch nicht beliebige Symbole gleichzeitig ausgewählt werden, sondern nur jene, die sich rechts und unterhalb des Cursors befinden. Anforderung 3 ist dennoch weitestgehend erfüllt.

Über andere Modi können auch lateinische Buchstaben oder die Buchstaben des Fingeralphabetes auf der Schreibfläche platziert werden. Diese können jedoch genauso wie die sonstigen GebärdenSchrift-Symbole einzeln verschoben werden. Einen Wortzusammenhang gibt es nicht.

SignWriter DOS verfügt über ein Gebärden-Wörterbuch, in dem verschriftlichte Gebärden unter einem lautsprachlichen Wort abgespeichert werden könnnen, um sie später in Texten wiederverwenden zu können. Anforderung 1 ist demnach erfüllt. Da Gebärden unter einem lautsprachlichen Begriff abgespeichert werden, kann ein gut geführtes Wörterbuch auch zur Übersetzung genutzt werden. Um eine Gebärde aus dem Wörterbuch in einen Text einzufügen, ist es jedoch notwendig, in einen Dialog zu wechseln, dort ein lautsprachliches Wort einzugeben, die Eingabe zu bestätigen, anschließend eine der gefundenen Gebärden auszuwählen und die Gebärde mit der `Enter`-Taste einzufügen. Entsteht ein GebärdenSchrift-Text nur durch Übersetzung eines lautsprachlichen Textes, geht durch das wiederholte Öffnen des beschriebenen Dialogs viel Zeit verloren. Dieser Prozess ließe sich eigentlich automatisieren. Anforderung 2 ist damit nur bedingt erfüllt.

Die Bedienung von SignWriter DOS erfolgt wie zuvor beschrieben ausschließlich über die Tastatur. Die Tastaturbelegung ist nicht selbsterklärend und wird nur in der beigefügten Anleitung erläutert. Wie zuvor beschrieben betrifft dies vor allem die Zuordnung der Tasten zu den Symbolgruppen, aber auch zu anderen Funktionen wie zum Beispiel dem Anpassen der einzelnen GebärdenSchrift-Symbole. Die Menüführung erfolgt über ein Menü am unteren Bildschirmrand (siehe Abbildung 13). Dort aufgelistete Menüpunkte lassen sich über die Taste `Alt` und den jeweiligen Anfangsbuchstaben des Menüpunktes auswählen. Dies ist innerhalb des Programms jedoch ebenfalls nicht ersichtlich. Anforderung 6 ist demnach nicht erfüllt. Dies ist auch der entscheidende Kritikpunkt an SignWriter DOS. Einsteiger müssen die Tastaturbelegung erst mühsam erlernen.

Die ausschließliche Bedienung über die Tastatur hat allerdings einen entscheidenden Vorteil: Kennt ein erfahrener Benutzer alle nötigen Tastaturbefehle, ist er in der Lage, Texte sehr zeiteffizient zu erstellen und zu bearbeiten. SignWriter DOS lässt sich dann fast so effizient bedienen wie ein gewöhnlicher Texteditor zum Erstellen lautsprachlicher Texte. Anforderung 7 ist damit sehr gut erfüllt.

3.2.2 SignWriter Java

Von SignWriter DOS existiert eine neuere Umsetzung in der Programmiersprache Java, genannt SignWriter Java[3]. Dank des der Programmiersprache Java inhärenten Konzepts der virtuellen Maschine läuft SignWriter Java auch unter verschiedenen Betriebssystemen. Zudem orientiert sich die Gestaltung der Oberfläche an aktuellen Standards, wie zum Beispiel eine Menüleiste mit Drop-Down-Menüs am oberen Bildschirm-/Fensterrand. Darüber hinaus verwendet SignWriter Java einen Symbolsatz in höherer Auflösung und erreicht somit eine bessere Grafikqualität als sein Vorgänger (siehe Abbildung 16).

Im Gegensatz zu SignWriter DOS ist leider keine deutsche Version des Programms verfügbar. Es werden bisher nur die Amerikanische (ASL) und die Nicaraguanische Gebärdensprache unterstützt. SignWriter Java bietet allerdings die Möglichkeit, Menüpunkte mit GebärdenSchrift-Symbolen zu bezeichnen (siehe Abbildung 15). Auf diese Weise soll Gehörlosen die Bedienung auch ohne Kenntnisse der jeweiligen Schriftsprache ermöglicht werden.

Bei SignWriter Java wurde versucht, das veraltete Bedienkonzept von SignWriter DOS mit Hilfe der moderneren Technik nahezu identisch umzusetzen. SignWriter Java verspielt dadurch viele Vorteile der moderneren Technologie und bei der Erstellung zeitgemäßer Benutzeroberflächen. Zum Beispiel ist die Bedienung von SignWriter Java fast genauso wenig selbsterklärend wie die des Vorgängers und erfüllt demnach nicht Anforderung 6. Lediglich die nun unterstützte Maussteuerung beim Selektieren und Verschieben von Symbolen verbessert die Anschaulichkeit ein wenig. Da die Umsetzung der Tastatursteuerung allerdings nicht vollständig und fehlerfrei gelang, lässt sich SignWriter Java trotz der moderneren Technologie effektiv schlechter bedienen als sein Vorgänger. Da die Bedienung größtenteils der von SignWriter DOS entspricht, nenne ich im Folgenden nur zwei Beispiele für eine Verschlechterung der Bedienung unter SignWriter Java.

[3] `http://www.signwriting.org/forums/software/sw50/index.html`

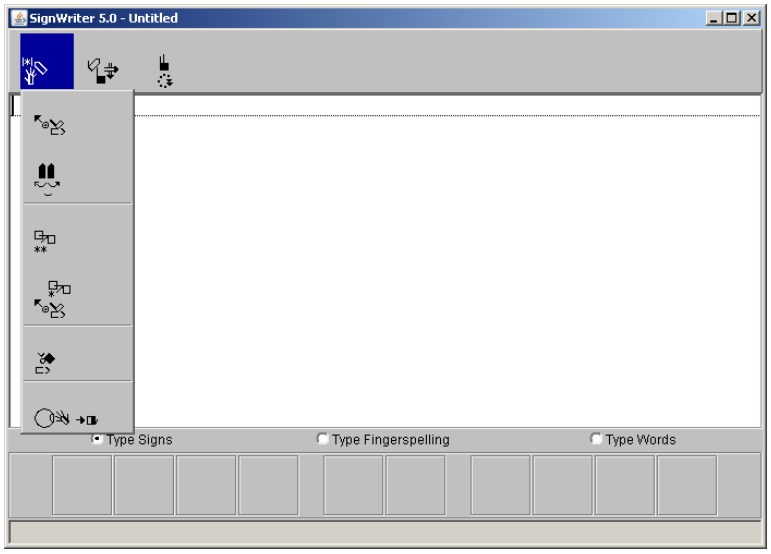


Abbildung 15: SignWriter Java unter Windows XP: Menüpunkte können mit Hilfe GebärdenSchrift benannt werden.

Das Bearbeiten und Erstellen von verschriftlichten Gebärden erfolgt nun modal in einer „Editierbox“, die über der Schreibfläche positioniert wird (siehe Abbildung 16). Erst nach einer Bestätigung mit der Taste `Enter` wird die so getippte oder veränderte Gebärde auf der Schreibfläche positioniert. Das Öffnen der Editierbox und die Arbeit darin verhalten sich dabei nicht immer fehlerfrei. Darunter leidet die Effizienz beim Erstellen von Texten.

Des Weiteren unterstützt SignWriter Java wie zuvor bereits erwähnt in Teilen die Bedienung mit der Maus. So lassen sich GebärdenSchrift-Symbole einer Gebärde innerhalb der Editierbox nun mit der Maus selektieren und per Drag & Drop verschieben. Jedoch leidet darunter stark die Effizienz, denn Gebärden und ihre Symbole lassen sich jetzt nicht mehr mit Hilfe der Tastatur selektieren. Dies unterbricht ebenfalls die ansonsten reibungslose und effiziente Steuerung unter SignWriter DOS. Anforderung 7 ist damit nicht mehr erfüllt.

SignWriter Java erbt größtenteils die Schwächen von SignWriter DOS und verschlechtert zudem die Effizienz der Steuerung, während sich die Anschaulichkeit nur leicht verbessert.

3.2.3 SignPuddle Online

SignPuddle Online[4] ist eine Internet-Platform zur GebärdenSchrift. Im Gegensatz zu den bisher vorgestellten Anwendungen dient SignPuddle nicht in erster Linie zum Schreiben von GebärdenSchrift-Texten, sondern vielmehr als Community-Platform zur gemeinschaftlichen Verwaltung und Erweiterung einer Gebärdendatenbank und zum Austausch

[4] `http://www.signbank.org/SignPuddle1.5/index.php?ui=8\&sgn=53`

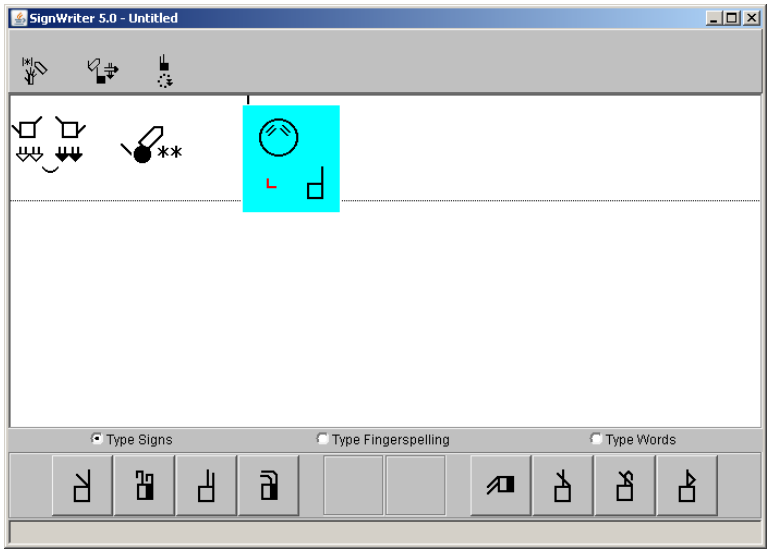


Abbildung 16: SignWriter Java unter Windows XP: Das Erstellen und Editieren von Gebärden erfolgt in einer Editierbox über der Schreibfläche (cyanfarben).

von GebärdenSchrift-Texten. Das Erstellen und Editieren von verschriftlichten Gebärden („SignMaker“), und die Erstellung von GebärdenSchrift-Texten bzw. das Übersetzen in GebärdenSchrift sind dabei in zwei verschiedene Anwendungen unterteilt. Ich beschreibe und bewerte deshalb beide Aspekte einzeln hintereinander. Auf die aus der Trennung entstehenden Nachteile gehe ich am Ende dieses Abschnittes ein.

Der in Abbildung 17 dargestellte SignMaker ermöglicht das Erstellen neuer Gebärden. Die einzelnen GebärdenSchrift-Symbole können per Drag & Drop aus einem Menü (siehe Abbildung 17 rechts) auf eine Zeichenfläche gezogen werden. Ebenso lassen sie sich dort verschieben und anordnen. Für die Sortierung und Anzeige der GebärdenSchrift-Symbole greift der SignMaker auf die in Abschnitt 2.3.2 beschriebene Datenbank des International SignWriting Alphabet zurück. Dabei werden jedoch sinnvollerweise als erstes die Gruppen und nicht die Kategorien angezeigt, um schneller auf die Symbole zugreifen zu können. Ansonsten präsentiert SignPuddle die komplette ungefilterte ISWA-Datenbank. Eine Filterung nach DGS-relevanten Symbolen findet also nicht statt. Eine Gruppe wird durch einen Klick auf das entsprechende Symbol geöffnet. Es ist dabei jedoch nicht erkennbar ob sich hinter einem angezeigten Symbol das Symbol selbst oder eine Kategorie verbirgt. Ebenso wenig wird angezeigt, in welcher Kategorie man sich gerade befindet. Dies und die ungefilterte Anzeige aller ISWA-Symbole verschlechtert die Bewertung im Hinblick auf Anforderung 5.

Eingefügte Symbole lassen sich über Buttons in den verschiedenen Ausprägungen anpassen, bei Handsymbolen also entsprechend der verschiedenen Handstellungen. Das heißt, sie können gedreht, gespiegelt und ihre Füllung verändert werden. Hinter diesen Anpassungen verbirgt sich eine Änderung der letzten drei Ziffern des in Abschnitt 2.3.2 beschrie-

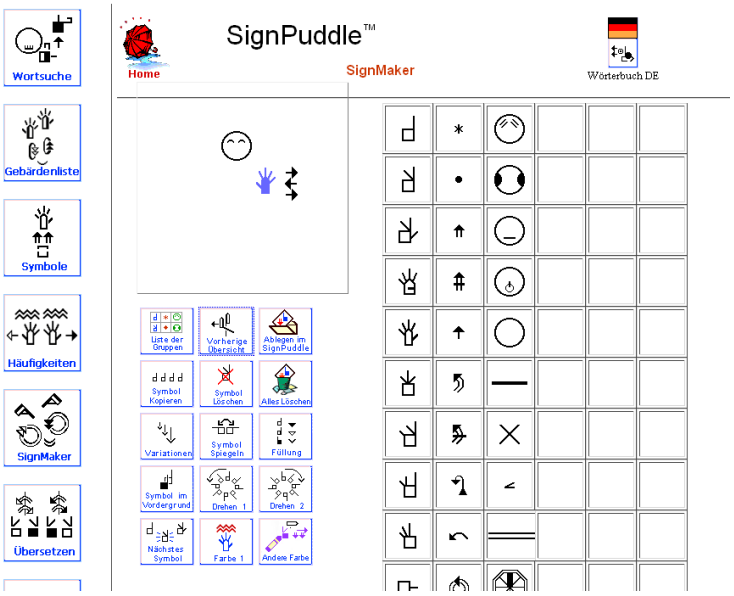


Abbildung 17: Der SignMaker von SignPuddle Online (Browser: Mozilla Firefox).

benen Identifikationscode der Symbole innerhalb des ISWA. Dies führt zwar dazu, dass sie je nach Symbolkategorie andere Auswirkungen haben, dennoch wurde zum Beispiel die Spiegelung als eigene Funktion ergänzt, die im Indentifikationscode nur indirekt durch eine Rotation abgebildet wird. Des Weiteren können Symbole über Buttons unter anderem kopiert, gelöscht oder in der Farbe angepasst werden. Anforderung 3 ist damit gut erfüllt. Die zuvor erwähnten Buttons sind jedoch nicht sinnvoll angeordnet (siehe Abbildung 17 Mitte). Dies verschlechtert die ansonsten anschauliche und selbsterklärende Steuerung ein wenig (Anforderung 6).

SignPuddle Online unterstützt als einziges vorgestelltes Programm direkt die Übersetzung von lautsprachlichen Texten in GebärdenSchrift. Diese funktioniert in drei modalen Dialogen. Der Benutzer gibt zunächst Text in ein Textfeld ein und bestätigt die Eingabe durch Drücken eines Buttons. Es ist auch möglich, Text aus externen Quellen zu kopieren und in das Textfeld einzufügen. Nach der zuvor genannten Bestätigung sucht SignPuddle Online für jedes Wort des eingegebenen Textes nach Übersetzungen in der Datenbank. Im nächsten Schritt muss der Benutzer für jedes der eingegebenen Wörter die passende Übersetzung unter den gefundenen Alternativen auswählen (siehe Abbildung 18). Anschließend muss er wieder bestätigen und der fertige Text in GebärdenSchrift wird angezeigt. Eine Flexionsauflösung findet bei der Auswertung der Eingabe nicht statt. Daher werden größtenteils nur unflektierte lautsprachliche Wörter erkannt.

In der Modalität der Dialoge liegt der größte Nachteil der Übersetzungsfunktion von SignPuddle. Der zuvor beschriebene Vorgang lässt sich nur komplett oder gar nicht durchführen. Will der Benutzer einen Tippfehler korrigieren, ist er gezwungen die ganze Proze-

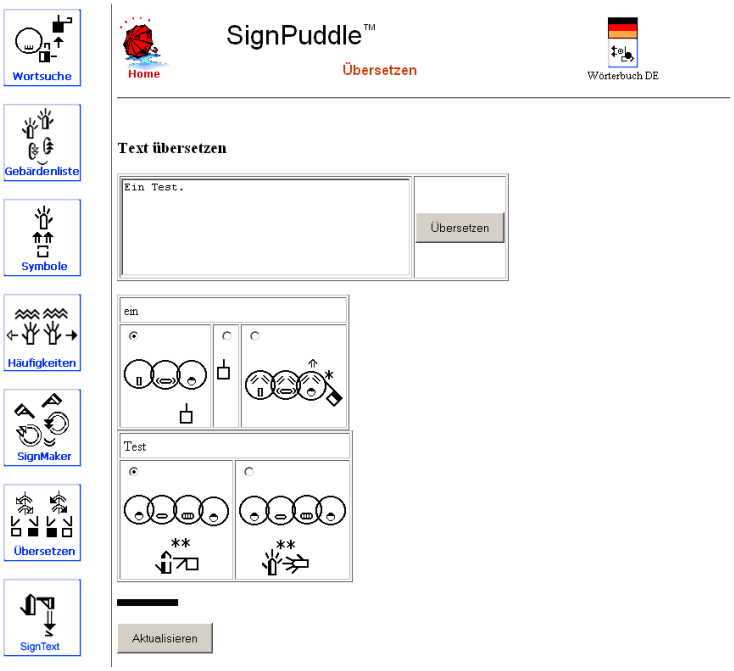


Abbildung 18: Der modale Dialog zum Auswählen der alternativen Übersetzungen bei der Übersetzung mit SignPuddle Online.

dur von vorne zu beginnen. Während die Übersetzungsfunktion für das Übersetzen kürzerer Sätze noch gut geeignet ist, wird das Erstellen von langen Texten durch die fehlende Möglichkeit zur Korrektur extrem mühsam bzw. nahezu unmöglich. Deshalb ist Anforderung 2 trotz der speziellen Übersetzungsfunktion nur unzureichend erfüllt.

Dies gilt auch, falls zu einem Wort keine Übersetzung gefunden wurde. Denn dann ist es notwendig, in den SignMaker zu wechseln, die Gebärde zu erstellen und die Übersetzung von vorne zu beginnen. Hierin zeigt sich der Nachteil der Trennung zwischen dem SignMaker und der Übersetzungsfunktion. Anforderung 4 ist demnach überhaupt nicht erfüllt.

3.2.4 Fazit

Die Tabelle in Abbildung 19 fasst die Bewertung der vorgestellten Programme im Hinblick auf die Anforderungen aus Abschnitt 3.1 zusammen. Keines der Programme erfüllt alle dort definierten Anforderungen. SignWriter DOS basiert auf einer veralteten Technik und verfügt über eine effiziente, jedoch in keiner Weise anschauliche Steuerung. SignWriter Java ist eine teilweise missglückte Umsetzung von SignWriter DOS und lässt sich schlechter bedienen als sein Vorgänger. SignPuddle Online ist vor allem als Community-Pattform gedacht und bietet eine einsteigerfreundliche, für erfahrene Benutzer jedoch ineffiziente

Steuerung. Aufgrund der Modalität bei der Übersetzung und der damit einhergehenden, fehlenden Möglichkeit zur Korrektur der Eingabe ist die Übersetzung längerer Texte mit SignPuddle zudem extrem mühsam.

		SignWriter DOS	SignWriter Java	SignPuddle Online
1.	Abspeichern und Einfügen vordefinierter verschriftlichter Gebärden	+	+	++
2.	Unterstützung bei der Übersetzung in LBG/DGS	-	-	-
3.	Erstellen und Editieren handschriftlicher Gebärden	+	+	++
4.	Editieren der Gebärden während der Texterstellung	++	++	- -
5.	Komfortabler Zugriff auf die GebärdenSchrift-Symbole	+	-	+
6.	Anschauliche und selbsterklärende Steuerung	- -	-	+
7.	Effiziente Steuerung	++	-	- -

Abbildung 19: Zusammenfassende Bewertung der existierenden GebärdenSchrift-Software.

In diesem Abschnitt habe ich existierende Anwendungen anhand der zuvor aufgestellten Anforderungen bewertet. Die dabei gewonnenen Analyseergebnisse sind in den im folgenden Kapitel beschriebenen Entwurf eingeflossen.

4 Entwurf des GebärdenSchrift-Editors

Auf der Grundlage der Ergebnisse des letzten Kapitels habe ich einen GebärdenSchrift-Editor entworfen, der in diesem Abschnitt der Arbeit vorgestellt wird. Der hier beschriebene Entwurf ist das Resultat eines agilen Entwicklungsprozesses, der in Abschnitt 5.1 näher beschrieben wird.

Wie in Abschnitt 1.1 beschrieben, ging dieser Arbeit ein studentisches Projekt voraus. Innerhalb dieses Projektes wurden bereits zwei Prototypen zur Erstellung von GebärdenSchrift-Texten entwickelt. Der eine Prototyp diente zur schnellen Übersetzung in LBG und DGS und funktionierte ähnlich wie der im nächsten Abschnitt beschriebene Übersetzungsmodus meines Entwurfs. Der zweite Prototyp diente zum Erstellen von verschriftlichten Gebärden aus einzelnen GebärdenSchrift-Symbolen. Aufgrund von Anforderung 4

aus Abschnitt 3.1 habe ich mich jedoch entschlossen, einen neuen GebärdenSchrift-Editor zu entwickeln, der beide Funktionen in sich vereint.

Eine weitere zentrale Anforderung aus Abschnitt 3.1 ist eine anschauliche und effiziente Steuerung. In diesem Entwurf habe ich mich entschieden, die Steuerung mit der Tastatur zu priorisieren. Dies geschieht vor allem aus Gründen der Effizienz. Ich möchte mit meinem Entwurf erreichen, dass sich das Programm wie ein klassischer Texteditor nahezu ausschließlich mit der Tastatur steuern lässt. Bei der Erstellung der Gebärden aus den Symbolen der GebärdenSchrift bot sich zunächst eine Maussteuerung nach den Prinzipien der direkten Manipulation an – vergleichbar mit der in Abschnitt 3.2.3 beschriebenen Steuerung bei der Erstellung von Gebärden auf SignPuddle Online. Der Benutzer „greift" Symbole mit dem Mauszeiger und bewegt diese per Drag & Drop aus einem Menü auf die Schreibfläche. Diese Steuerung ist anschaulich jedoch meiner Meinung nach nicht effizient. Da zum Tippen von lautsprachlichem Text ansonsten die Tastatur zwingend benötigt wird, müsste der Benutzer des Öfteren zwischen der Maus und der Tastatur wechseln, was auf Dauer Zeit und Anstrengung kosten würde. Ein klassischer Texteditor wäre zudem höchst unpraktisch, wenn sich einzelne Buchstaben nur mit der Maus auf die Schreibfläche ziehen ließen. Nichtsdestotrotz ist es entscheidend, die Tastaturbelegung also die Zuordnung der Tasten zu den Funktionen des Programms anschaulich zu visualisieren, um der Anforderung einer selbsterklärenden Steuerung gerecht zu werden. Dies geschieht mit Hilfe des Tastaturfeldes, das als zentrales Element dieses Entwurfs in Abschnitt 4.3 genauer beschrieben wird.

Die Gesamtfunktionalität dieses Entwurf lässt sich in zwei Modi aufteilen: der Übersetzungsmodus und der Bearbeitungsmodus. Diese werden in den folgenden beiden Abschnitten näher beschrieben.

4.1 Der Übersetzungsmodus

Der in Abbildung 20 gezeigte Übersetzungsmodus dient dazu, GebärdenSchrift schnell anhand der lautsprachlichen Bedeutung zu tippen bzw. lautsprachliche Texte schnell in LBG zu übersetzen. Oberhalb der Seite wird dazu ein klassisches Textfeld angezeigt. Gibt der Benutzer dort Text ein, wird dieser sofort in einzelne Wörter zerlegt. Für diese wird automatisch in einer Datenbank nach passenden vordefinierten Gebärden gesucht. Letztere werden unterhalb des Textfeldes auf der Seite angezeigt. Dort erscheint also quasi die LBG-Übersetzung des eingegebenen Satzes. Die Eingabe von Text funktioniert dabei wie von einem Texteditor gewohnt. Es ist zum Beispiel möglich, Text zu markieren, zu kopieren und einzufügen. Dies erlaubt es auch, Text aus externen Quellen wie zum Beispiel einer Internetseite in die Zwischenablage zu kopieren und im GebärdenSchrift-Editor einzufügen. Dieser wird dann ebenso wie eingetippter Text übersetzt.

Wie in Abschnitt 2.2 beschrieben, lässt sich die Übersetzung jedoch nicht vollständig automatisieren. Für ein und dasselbe Wort können verschiedene kontextabhängige Übersetzungen existieren. Dies wird durch eine Anzeige unterhalb der Gebärden angezeigt. Dort steht, wie viele alternative Übersetzungen gefunden wurden und welche davon gera-

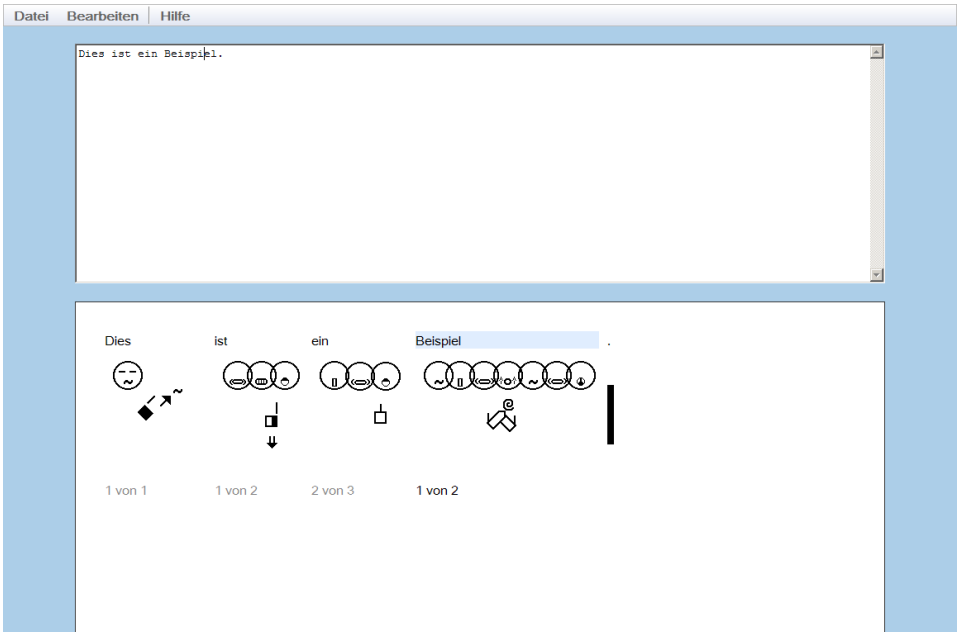


Abbildung 20: Der Übersetzungsmodus: Der eingetippte Text (oben) wird direkt in GebärdenSchrift übersetzt (unten).

de angezeigt wird. Der Benutzer kann mit Hilfe der Tastatur (vorerst `Bild hoch` und `Bild runter`) die verschiedenen Übersetzungen für die selektierte Gebärde durchschalten. Dies ist immer die Gebärde, die zu dem Wort im Textfeld gehört, in dem sich der Cursor gerade befindet. Die selektierte Gebärde wird zusätzlich farblich gekennzeichnet.

Das Erstellen von DGS-Texten erfolgt analog zur Erstellung von LBG-Texten. Der Benutzer tippt die Bedeutungen der Gebärden ein, die er für den Text benötigt. Er muss dabei jedoch schon die Syntax der DGS beachten. Der Benutzer tippt also in diesem Fall keinen korrekten lautsprachlichen Text, sondern eine mit der Glossentranskription vergleichbare Repräsentation des DGS-Textes in der Lautsprache.

Langfristig soll das Textfeld vor dem Benutzer verborgen werden, sodass er scheinbar direkt auf der Seite tippt. Auf diese Weise wird der Bezug zwischen dem getippten lautsprachlichen Text und der Repräsentation in GebärdenSchrift noch deutlicher. Zunächst stellt das Textfeld aber Funktionalitäten wie zum Beispiel Copy & Paste bereit.

4.2 Der Bearbeitungsmodus

Der in Abbildung 21 dargestellte Bearbeitungsmodus dient zum Erstellen und Editieren der verschriftlichten Gebärden. Drückt der Benutzer eine geeignete Taste, wechselt das

Programm in den Bearbeitungsmodus, der sich durch erneutes Drücken derselben Taste oder mit zum Beispiel der `Esc`-Taste wieder verlassen lässt. Statt des Wortes oberhalb der selektierten Gebärde wird nun die Gebärde selbst farblich hinterlegt, um dem Benutzer zu zeigen, dass sie sich nun bearbeiten lässt. An Stelle des Textfeldes wird das zuvor beschriebene Tastaturfeld angezeigt, das die Tastaturbelegung anschaulich darstellt. Dessen Aufbau und Verhalten beim Einfügen neuer GebärdenSchrift-Symbole wird wie zuvor erwähnt in Abschnitt 4.3 ausführlich beschrieben.

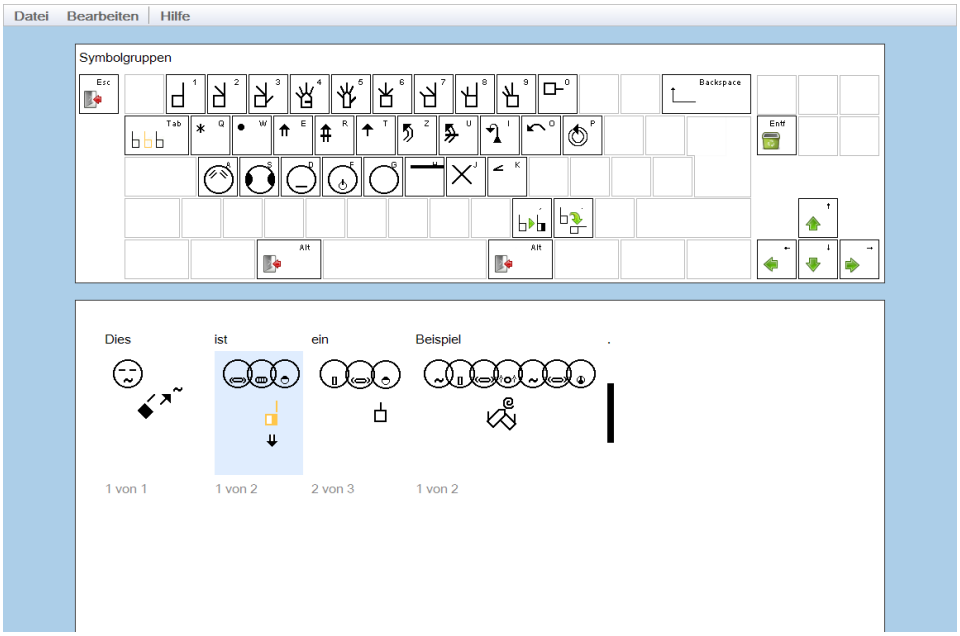


Abbildung 21: Der Bearbeitungsmodus mit dem Tastaturfeld (oben). Das orange gefärbte Symbol (unten) ist selektiert und kann manipuliert werden.

Darüber hinaus lassen sich im Bearbeitungsmodus die einzelnen Symbole der Gebärde mit einer Taste nacheinander selektieren. Ein selektiertes Symbol wird entsprechend farblich hervorgehoben. Selektierte Symbole können mit Hilfe der Tastatur verschoben oder entfernt werden. Handelt es sich um Handsymbole, können diese mit Hilfe weiterer geeigneter Tasten entsprechend der verschiedenen Handstellungen gedreht bzw. angepasst werden.

4.3 Einfügen von GebärdenSchrift-Symbolen mit dem Tastaturfeld

Ein zentrales Element meines Entwurfes ist eine selbsterklärende Tastatursteuerung, die Effizienz beim Einfügen und Editieren von GebärdenSchrift-Symbolen mit einer anschaulichen Visualisierung vereint. Sie soll es ermöglichen, GebärdenSchrift-Symbole wie die

Buchstaben des lateinischen Alphabetes zu „tippen“. Die Steuerung orientiert sich dabei teilweise an SignWriter DOS. Obwohl SignWriter DOS technologisch veraltet ist, lassen sich mit keinem der anderen in Abschnitt 3.2 vorgestellten Programme GebärdenSchrift-Texte so effizient erstellen. Wie in Abschnitt 3.2.1 beschrieben ist die Steuerung von SignWriter DOS jedoch nicht selbsterklärend. Deshalb habe ich in meinem Entwurf darauf wert gelegt, die Tastaturbelegung zu visualisieren. Um den Einstieg in die Software zu erleichtern, wird diese auf einem virtuellen Abbild der Tastatur grafisch dargestellt.

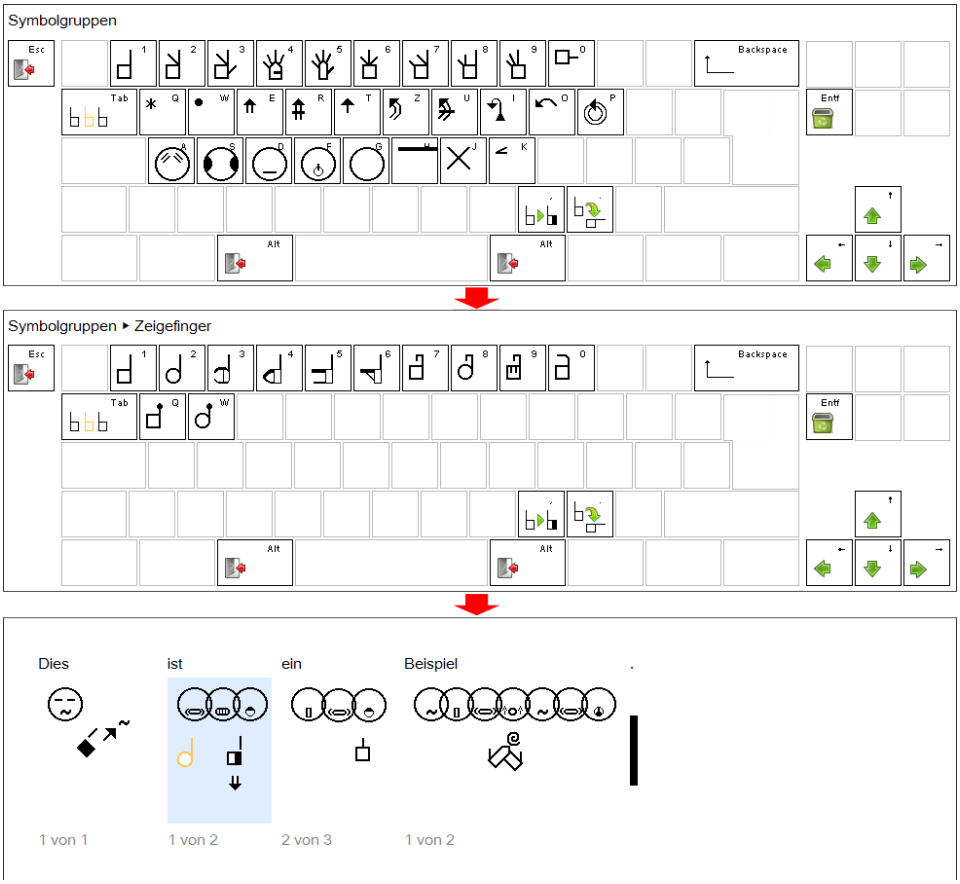


Abbildung 22: Die Funktionsweise des Tastaturfeldes: Der Benutzer drückt zunächst die Taste „1“ und betritt die Kategorie „Zeigefinger“. Anschließend fügt er das zweite Symbol mit der Taste „2“ ein.

Wie in allen vorgestellten GebärdenSchrift-Anwendungen sind die zahlreichen GebärdenSchrift-Symbole aufgrund ihrer großen Anzahl in Kategorien und Unterkategorien aufgeteilt. Ich verwende für den GebärdenSchrift-Editor die Aufteilung gemäß des International SignWriting Alphabets. Die Aufteilung der Symbole in Kategorien kann als eine Art Baumstruktur angesehen werden. Dabei stehen die Blätter des Baumes für die Symbole, die restlichen Knoten repräsentierten die in Abschnitt 2.3.2 beschriebenen Symbolkatego-

rien und -gruppen. Ich habe mich jedoch für eine möglichst flache Hierarchie entschieden, damit zum Einfügen eines Symbols nicht zu viele Kategorien durchlaufen werden müssen. Daher stellt die oberste Ebene des Baumes bereits die in Abschnitt 2.3.2 beschriebenen 30 Symbolgruppen dar, die darunter liegende enthält bereits die einzelnen Symbole.

Die Symbolgruppen werden, repräsentiert durch ein Symbol, beim Wechsel in den Bearbeitungsmodus auf die Tasten des Tastaturfeldes verteilt. Dies bedeutet, ein Symbol erscheint auf jeder „virtuellen“ Taste. Diese Abbilder der Tasten sind zusätzlich mit dem entsprechenden Buchstaben gekennzeichnet, der auf der physischen Tastatur abgebildet ist. Auf diese Weise wird direkt der Bezug zwischen der Taste auf der Tastatur und dem entsprechenden Symbol hergestellt. Nicht verwendete Tasten werden ausgegraut. Drückt der Benutzer nun eine der Tasten, betritt er die entsprechende Symbolgruppe. Nun werden die zu der Gruppe gehörigen Symbole auf die Tasten verteilt. Ein weiterer Druck auf eine der Tasten fügt das entsprechende Symbol zur selektierten Gebärde hinzu. Über eine Taste (zum Beispiel `Backspace`) lässt sich die aktuelle Symbolkategorie immer wieder verlassen, um zur nächst höheren zu gelangen. Eine Navigationsleiste am oberen Rand des Tastaturfeldes zeigt ähnlich wie in einem Dateibrowser den Pfad zur aktuellen Position im Symbolbaum, also die Namen der übergeordneten Symbolgruppen. Abbildung 22 zeigt den Vorgang des Eingügens anhand eines Beispiels.

Ebenso sind auch alle weiteren im Bearbeitungsmodus verwendeten Tasten auf dem Tastaturfeld dargestellt und mit entsprechenden Symbolen versehen, die ihre Funktion veranschaulichen. Dies betrifft die Tasten zum Selektieren, Verschieben, Anpassen oder Entfernen der GebärdenSchrift-Symbole.

5 Softwaretechnische Umsetzung

Dieses Kapitel beschreibt die softwaretechnische Umsetzung des im vorherigen Kapitel vorgestellten Entwurfs für den GebärdenSchrift-Editor. Nach einer Beschreibung der verwendeten Methode bei der Projektplanung stelle ich kurz das Google Web Toolkit vor, das als zentrale Technologie bei der Implementierung des Entwurfs verwendet wurde. Anschließend erläutere ich die Architektur des GebärdenSchrift-Editors, aufgeteilt in die drei Komponenten Werkzeug, Material und Webservice.

5.1 Verwendete Methode

Dieser Abschnitt beschreibt, wie bei der Planung und Entwicklung des in Kapitel 4 beschriebenen Entwurfs vorgegangen wurde.

Die verwendete Vorgehensweise ist dabei an die Prinzipien und Methoden der Agilen Softwareentwicklung angelehnt. Der Entwurf wurde dazu in Stories aufgeteilt, die jeweils einen Aspekt seiner Funktionalität detailliert beschrieben haben. Zur Visualisierung dieser Stories habe ich ein Mock-Up gestaltet, das den Aufbau und die Funktionalität der Benutzeroberfläche veranschaulicht hat (siehe Anhang B). Der Entwurf wurde anhand

dieses Mock-Ups mit Stefan Wöhrmann abgestimmt. Zur detaillierteren Planung wurden die Stories dann in Tasks verfeinert und mit einer relativen Aufwandsschätzung versehen. Anschließend habe ich die Stories priorisiert und aufgrund dessen in Iterationen eingeteilt. Dabei sollte zunächst die Übersetzungsfunktionalität des zu Beginn von Abschnitt 4 erwähnten früheren Prototypen wiederhergestellt werden. Weiterhin war die Umsetzung des Übersetzungs- und Bearbeitungsmodus und die damit verbundene unterschiedliche Behandlung der Tastatureingaben entscheidend für die Architektur der Anwendung. Als zentrales Element des Entwurfs sollte zudem das Einfügen von Symbolen mit dem Tastaturfeld möglichst früh realisiert und erprobt werden, um Risiken zu minimieren. Für die Einteilung der Iterationen war weiterhin entscheidend, dass jede Iteration einen lauffähigen Prototypen mit einem Mehrwert zum Ergebnis hat. Die resultierende Planung und detailliertere Begründungen für die Priorisierung der Stories kann in Anhang A nachgelesen werden.

Die Planung wurde während der Entwicklung teilweise ergänzt und angepasst. Bis auf die letzte Iteration wurden alle Tasks umgesetzt.

5.2 Verwendete Technologien

Ich habe mich entschlossen, den GebärdenSchrift-Editor als browserfähige Internetanwendung auf Basis des Google Web Toolkits zu realisieren. Dies gewährleistet Plattformunabhängigkeit und hat zudem den Vorteil, dass für seine Verwendung neben einem Browser keine Software installiert werden muss. Darüber hinaus ist es so möglich, den GebärdenSchrift-Editor langfrisitg in eine mit SignPuddle (vgl. Abschnitt 3.2.3) vergleichbare Community-Plattform einzubinden. Im Folgenden gebe ich einen kurzen Überblick über das Google Web Toolkit.

5.2.1 Das Google Web Toolkit

Das Google Web Toolkit (GWT) ist ein Open-Source-Framework für die Entwicklung von AJAX-basierten Webanwendungen, das 2006 von Google veröffentlicht wurde. AJAX steht für „Asynchronous JavaScript and XML“. Dieses Konzept ermöglicht es, HTTP-Anfragen durchzuführen, während eine HTML-Seite angezeigt wird, ohne diese komplett neu zu laden. Auf diese Weise lassen sich nur Teile der HTML-Seite oder reine Nutzdaten im Hintergrund nachladen. AJAX ist damit das grundlegende Konzept für die Entwicklung interaktiver Webanwendungen.

Im Gegensatz zu anderen Frameworks zur Entwicklung von AJAX-basierten Webanwendungen erlaubt es GWT, sowohl den Server-Code als auch den Client-Code in Java zu schreiben. Ein Compiler übersetzt letzteren anschließend im Hintergrund in optimiertes Java-Script und HTML, sodass mit GWT entwickelte Anwendungen im Browser ohne PlugIn ausgeführt werden können. Dabei werden jedoch nur einige Klassen und Interfaces aus den Packages `java.lang`, `java.util` und `java.io` unterstützt, darunter zum Beispiel die gebräuchlichsten `Collections`. Zusätzlich besteht die Möglichkeit,

Methoden direkt in Java-Script zu programmieren. Bei der Übersetzung in Java-Script passt GWT den Code für die Eigenheiten verschiedener Browsertypen an, sodass GWT-Programme in den meisten Fällen mit den verschiedenen gängigen Browsern kompatibel sind.

Des Weiteren beinhaltet GWT eine mit SWING vergleichbare Widgit-Bibliothek. Auf diese Weise lässt sich die Oberfläche einer Webanwendung ähnlich programmieren wie die einer reinen Java Anwendung. Wie in SWING generieren Widgits wie Buttons Events, die entsprechende Listener über Zustandsänderungen oder Aktivitäten des Benutzers informieren. Dabei greift GWT soweit möglich auf native HTML-Elemente wie zum Beispiel Buttons, Tabellen oder Textfelder zurück. Das Aussehen der Widgits lässt sich deshalb über Cascading Style Sheets (CSS) individualisieren.

Für die Kommunikation zwischen der Clientanwendung und einem Java-Servlet bietet GWT eine Unterstützung für Remote Procedure Calls (RPC). Das heißt, in der Clientanwendungen können Methoden des in Java geschriebenen Servlets verwendet werden. Die entsprechende HTTP-Kommunikation erfolgt automatisch im Hintergrund. Auf der Client-Seite muss dazu beim Methodenaufruf zusätzlich ein Callback-Objekt erzeugt werden, dessen Methoden die Reaktion des Servlets behandeln.

Eine weitere Besonderheit von GWT ist der Hosted Mode. Mit GWT entwickelte Anwendungen lassen sich testweise im Hosted Mode starten. GWT startet dann die Anwendung in einem simulierten Browser. Während im Hintergrund die Java-Anwendung läuft, wird das Verhalten nach dem Übersetzen in Java-Script simuliert. Auf diese Weise lässt sich die Client-Anwendung einfach testen und trotzdem wie eine reine Java Anwendung debuggen. Die Entwicklung in der Hochsprache Java bringt weitere Vorteile: GWT unterstützt automatisierte Tests mit JUnit und stellt ein PlugIn für die Entwicklungsumgebung Eclipse bereit. (Vgl. Gup08)

5.3 Architektur

Dieser Abschnitt beschreibt die Architektur des entwickelten GebärdenSchrift-Editors. Ich erläutere die Verteilung verschiedener Funktionen auf Programmkomponenten und deren Zusammenspiel.

Beim Programmieren der Anwendung stand für mich die Trennung der Fachlichkeit von der Darstellung bzw. deren Manipulation im Vordergrund. Abbildung 23 zeigt dementsprechend, wie sich die Anwendung auf oberster Ebene in die zwei Komponenten Werkzeug und Material aufteilen lässt. Das Werkzeug des GebärdenSchrift-Editors kommuniziert zusätzlich mit dem Webservice. Die Benennung der Komponenten Werkzeug und Material ist an die gleichlautenden Konzepte im Werkzeug- und Materialansatz angelehnt (vgl. Z$^+$98). Dieser Ansatz orientiert sich an einer Fabrik oder einem Handwerksbetrieb, in dem Mitarbeiter Materialien mit Werkzeugen bearbeiten. Das Software-Material beinhaltet dabei diejenigen Klassen, die Daten oder Gegenstände modellieren, die ein Benutzer im Rahmen seiner Aufgabe bearbeitet oder erstellt. Das Software-Werkzeug stellt Methoden zur Bearbeitung des Materials bereit und automatisiert Arbeitsschritte. Diese Aufteilung ist

mit dem Model-View-Controller-Entwurfsmuster vergleichbar. In diesem Fall sind View und Controller zum Werkzeug zusammengefasst.

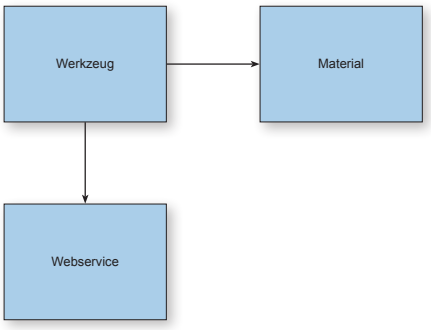


Abbildung 23: Die Architektur des GebärdenSchrift-Editors lässt sich in die Komponenten Werkzeug, Material und Webservice unterteilen.

In den folgenden beiden Abschnitten beschreibe ich die beiden Komponenten Werkzeug, Material und Webservice genauer. Dabei werden Klassennamen durch einen äquidistanten Schrifttyp verdeutlicht, zum Beispiel `KlassenName`.

5.3.1 Material

Das Material des GebärdenSchrift-Editors lässt sich in zwei Komponenten aufteilen.

Da wären zunächst diejenigen Klassen, welche die Verarbeitung des Textes betreffen. Abbildung 24 stellt diese Klassen in einem Klassendiagramm dar. Analog zu Textverarbeitungsprogrammen arbeitet der Benutzer an einem Dokument, in diesem Fall das `SignWritingDocument`. Ein `SignWritingDocument` verfügt neben einem Namen über eine `Region` und einen `Author`. Die Angabe der Region soll dem in Abschnitt 2.1 beschriebenen Problem der starken regionalen Unterschiede in der DGS entgegenwirken und kann zum Beispiel genutzt werden, um die Resultate bei der Übersetzung dem regionalen Vokabular entsprechend zu filtern.

In dem zu Beginn von Abschnitt 4 erwähnten Übersetzerprototypen bestand noch die Möglichkeit mehrere GebärdenSchrift-Texte unter einem Dokument zu bearbeiten, um zum Beispiel einen LBG- und einen DGS-Text gegenüber zustellen. Diese Funktionalität ist in dem von mir entwickelten GebärdenSchrift-Editor noch nicht umgesetzt. Das Material wurde jedoch teilweise übernommen und beinhaltet diese Möglichkeit schon. Daher besteht ein `SignWritingDocument` aus mehreren Texten, repräsentiert durch die Klasse `Text`. Diese wiederum bestehen aus `Tokens`. Ein `Token` ist entweder ein `Word` oder ein `SpecialCharacter`. Diese Unterscheidung ist notwendig, weil Wörter und Sonderzeichen unterschiedlich übersetzt und dargestellt werden.

Die zweite Komponente des Materials modelliert die GebärdenSchrift. Sie wird in Abbildung 25 durch ein Klassendiagramm dargestellt. Die Klasse `Sign` repräsentiert eine

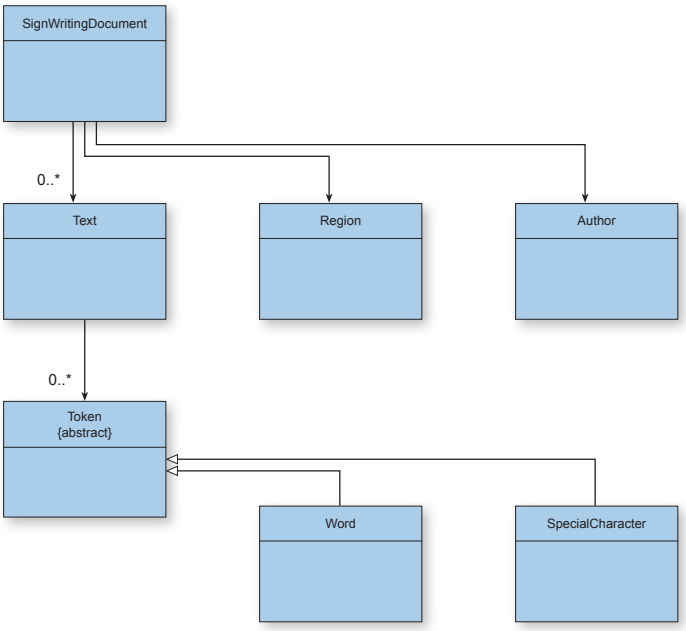


Abbildung 24: Die Material-Komponente SignWiritingDocument.

mit der GebärdenSchrift verschriftlichte Gebärde. Ein `Sign` hat mehrere (lautsprachliche) Bedeutungen und eine `Region`, in der sie gebräuchlich ist. Darüber hinaus besteht sie aus mehreren GebärdenSchrift-Symbolen mit bestimmten Koordinaten, analog zur in Abschnitt 2.3 beschriebenen, zweidimensonalen Anordnung der Symbole zu Bildung von verschriftlichen Gebärden. Ein Symbol der GebärdenSchrift wird durch die Klasse `Symbol` modelliert. Da Symbole in der Anwendung zum Teil auch ohne Position auftreten (zum Beispiel innerhalb des im nächsten Abschnitt beschriebenen `KeyboardPanels`), ist die Position in der zusätzlichen Klasse `SymbolWithPosition` gekapselt. Ein `Symbol` verfügt neben einem Namen über einen `SymbolCode`. Dieser modelliert den in Abschnitt 2.3.2 beschriebenen Identifikationscode der GebärdenSchrift-Symbole innerhalb des International SignWriting Alphabet und wird zur Generierung der URL verwendet, mit der das Bild zum Symbol aus der ISWA-Datenbank geladen wird.

Bei der Übersetzung wird nun jedes `Word` auf eine Liste von `Signs` abgebildet und jeder `SpecialCharacter` auf ein `Symbol`. Diese Zuordnung geschieht über den Webservice bzw. ein entsprechendes Servlet. Dies wird in Abschnitt 5.3.3 genauer beschrieben.

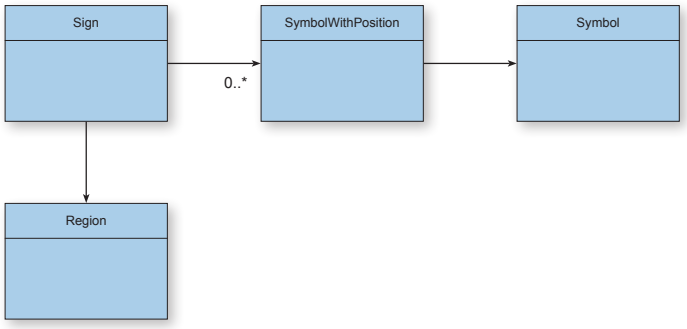


Abbildung 25: Die Material-Komponente Sign.

5.3.2 Werkzeug

Das Werkzeug also der GebärdenSchrift-Editor lässt sich vereinfacht mit dem in Abbildung 26 gezeigten Klassendiagramm beschreiben. Diese Darstellung ist insofern vereinfacht, als sie nicht alle Klassen enthält, sondern nur diejenigen, die für die Funktionalität entscheidend sind.

Die zentrale Funktionalität steckt dabei in der Klasse `SignWritingEditor`. Diese Klasse stellt den `EntryPoint` der GWT-Client-Anwendung dar. Das heißt, beim Start der Anwendung wird ein `SignWritingEditor`-Objekt erzeugt.Sie hält die Referenz auf das zuvor beschriebene `SignWritingDocument`-Material, auf die im nächsten Abschnitt beschriebenen Servlets und auf die Objekte der Benutzeroberfläche. Letztere wird durch die Klasse `MainPanel` repräsentiert. Ein entsprechendes Objekt hält die Referenz auf die beiden Eingabekomponenten `KeyboardPanel` und `TextArea` und verwaltet deren Sichtbarkeit und Fokussierung im Übersetzungs- und im Bearbeitungsmodus. Sie stellen die technische Umsetzung des in Kapitel 4 beschriebenen Text- und des Tastaturfeldes dar. Darüber hinaus beinhaltet das `MainPanel` das visuell seitenähnliche Panel, in dem die `TokenBox`-Objekte angzeigt werden. Eine `TokenBox` ist die UI-Repräsentation eines zuvor beschriebenen `Tokens` im `SignWritingDocument`-Material und zeigt das passende GebärdenSchrift-Symbol bzw. die passende verschriftlichte Gebärde an. Die Klasse `TokenBox` steht dabei analog zum Material in einer Vererbungsbeziehung zu den Klassen `WordTokenBox` und `SpecialCharacterTokenBox`. Diese Unterscheidung ist notwendig, da ein Sonderzeichen bzw. das entsprechende GebärdenSchrift-Symbol nicht bearbeitet werden können – im Gegensatz zu einem Wort bzw. den entsprechenden Gebärden.

Nach dem eingangs erwähnten Model-View-Controller-Entwurfsmuster wäre die `SignWritingEditor`-Klasse die Controller-Komponente, während die restlichen Klassen aus Abbildung 26 der View zuzuordnen wären. Teilweise beinhaltet die Benutzeroberfläche jedoch auch Controller-Funktionen, so zum Beispiel die `WordTokenBox`, die das von ihr angezeigte `Sign` im Bearbeitungsmodus manipuliert.

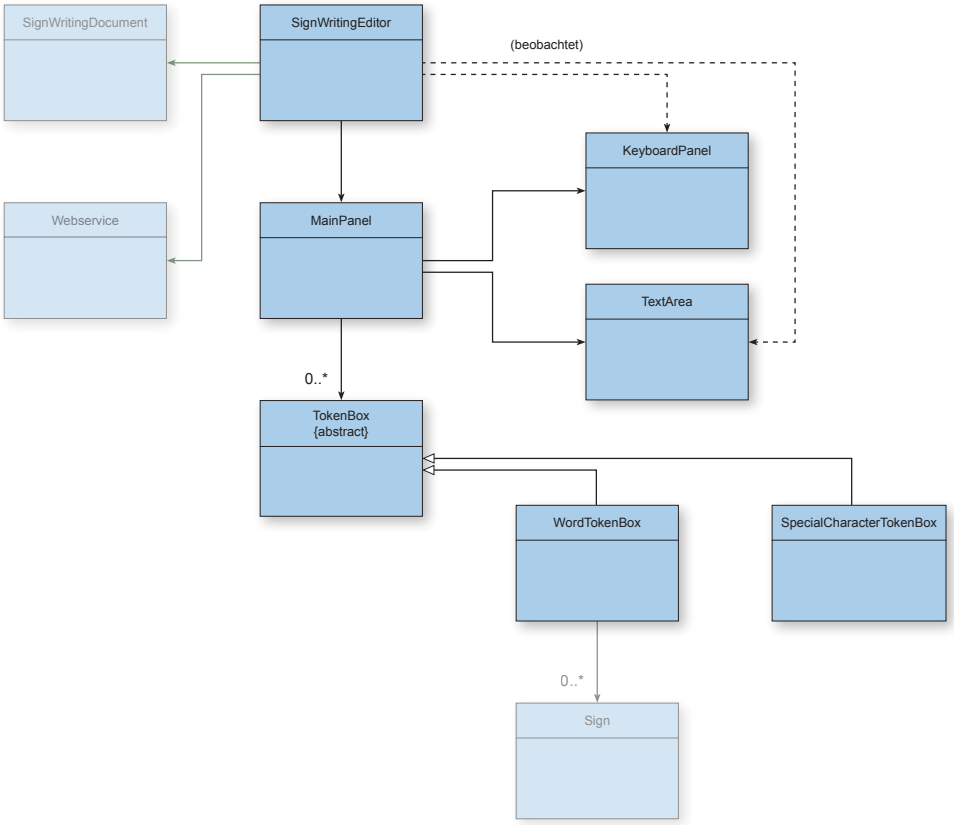


Abbildung 26: Die zentralen Klassen innerhalb der Werkzeug-Komponente (zusätzlich angedeutet: Referenzen auf die Materialien und den Webservice).

Die Funktionalität im Übersetzungs- und im Bearbeitungsmodus lässt sich wie folgt zusammenfassen:

- Übersetzungsmodus: Gibt der Benutzer lautsprachlichen Text in die `TextArea` ein, werden Listener innerhalb des `SignWritingEditor`-Objekts benachrichtigt. Dort wird der Text aus der `TextArea` von einem `Tokenizer` in `Tokens` zerlegt. Mit Hilfe dieser `Tokens` wird zum einen das `SignWritingDocument`-Material aktualisiert. Zum anderen wird für jedes `Token` eine entsprechende `TokenBox` erstellt, diese mit Hilfe der Servlets mit `Signs` bzw. `Symbols` bestückt und dem `MainPanel`-Objekt zur Anzeige übergeben. Dabei wird aus Effizienzgründen versucht, existierende `TokenBoxes` soweit wie möglich wieder zu verwenden.
- Bearbeitungsmodus: Drückt der Benutzer eine Taste, werden genauso wie im Übersetzungsmodus Listener innerhalb des `SignWritingEditor`-Objekts benach-

richtigt. Sofern es sich um eine Taste zu Manipulation von selektierten Symbolen handelt, werden über das `MainPanel` Methoden am selektierten `TokenBox`-Objekt aufgerufen. Diese nehmen dann die Änderungen an der dargestellten Gebärde bzw. am entsprechenden `Sign`-Objekt vor, zum Beispiel die Rotation oder das Verschieben von Symbolen. Die Rotation und die Änderung der Füllung ist durch eine Anpassung der letzten beiden Ziffern des `SymbolCode` realisiert (vgl. Abschnitt 2.3.2). Handelt es sich bei der gedrückten Taste nicht um eine der Tasten zur Symbolmanipulation, wird das `KeyboardPanel` benachrichtigt. Dieses gibt zurück ob die Taste zu einem Symbol gehört und um welches es sich handelt und aktualisiert sich entsprechend (falls eine Symbolgruppe betreten wurde). Ein so erhaltenes Symbol wird erneut über das `MainPanel` und die selektierte `TokenBox` zum dargestellten `Sign`-Objekt hinzugefügt.

Während die zuvor erwähnte `TextArea` ein GWT-Standard-Widgit ist, verbirgt sich hinter dem `KeyboardPanel` ein selbst erstelltes Widgit. Wie in Kapitel 4 beschrieben ist das Tastaturfeld eine zentrale Komponente meines Entwurfs für den GebärdenSchrift-Editor. Aus diesem Grund beschreibe ich dessen Implementation, also das `KeyboardPanel`, an dieser Stelle genauer.

Wie in Abschnitt 4.3 beschrieben stellt das `KeyboardPanel` die Tasten der Tastatur grafisch dar und visualisiert die Verteilung der Programmfunktionen bzw. der Symbole auf diese Tasten. Die grafische Repräsentation einer Taste wird durch die Klasse `KeyPanel` implementiert. Diesen kann jeweils ein Symbol in Form einer URL zu einem entsprechenden Bild zugewiesen werden. Zudem ist es möglich, die Taste als belegt oder unbenutzt, also ausgegraut, zu kennzeichnen. Die darzustellenden GebärdenSchrift-Symbole werden dem `KeyboardPanel` in Form einer Datenstruktur übergeben, die sie in Kategorien ordnet. Diese Datenstruktur ist analog zur in Abschnitt 2.3.2 beschriebenen Kategorisierung der GebärdenSchrift-Symbole vergleichbar mit einem Baum. Wie in Abschnitt 4.3 erläutert verwende ich in der Anwendung jedoch nur eine Aufteilung in die ISWA-Gruppen, obwohl die Datenstruktur und dessen Anzeige durch das `KeyboardPanel` theoretisch einen Baum mit beliebiger Tiefe erlaubt. Beim Wechsel in den Bearbeitungsmodus werden zunächst die Gruppen und ihre Symbole auf die `KeyPanels` verteilt. Welche Tasten dafür verwendet werden, ist in die Konfigurationsklasse `KeyboardConstants` ausgelagert. Dort werden an zentraler Stelle alle Tastenbelegungen vermerkt. Neben den Tasten zur Manipulation von Symbolen ist dort in einem Array festgelegt, welche Tasten zum Einfügen der Symbole verwendet werden. Das `KeyboardPanel` richtet sich nach dieser Konfiguration und graut dort nicht verwendete Tasten wie zuvor beschrieben aus. Zusätzlich enthält das `KeyboardPanel` ein `NavigationLabel`, das die Implementation der in Abschnitt 4.3 erläuterten Navigationsleiste darstellt. Es zeigt den Pfad zur aktuell angezeigten Symbolgruppe/-kategorie.

Ähnlich wie die Konfiguration der Tasten sind auch andere Konstanten innerhalb des Werkzeugs in die Klassen `KeyboardConstants`, `UIConstants` und `NetworkConstants` ausgelagert. Auf diese Weise lassen sich neben der Tastaturbelegung Parameter der Bedienoberfläche, wie zum Beispiel die Breite des seitenähnlichen Panels oder die URL zu den Symbolen an zentraler Stelle konfigurieren.

5.3.3 Webservice

Der vorherige Abschnitt hat das Werkzeug der entwickelten Anwendung beschrieben. Wie bereits angedeutet ist dieses darauf ausgelegt, dass Daten wie die verschriftlichten Gebärden und die verfügbaren GebärdenSchrift-Symbole von einem zentralen Server geladen und durch das Werkzeug nur angezeigt werden. Wie zu Beginn von Abschnitt 4 kurz erwähnt baut der entwickelte GebärdenSchrift-Editor teilweise auf zwei Prototypen auf. Diese verfügten über einen gemeinsamen Webservice, der dei zuvor beschriebenen Daten zum Teil bereit gestellt hat. Er war in der Lage, Datenbankanfragen per HTTP entgegenzunehmen und die entsprechenden Ergebnisse in Form von JSON-Objekten zurückzuliefern. Für die Übersetzung war eine Anfrage besonders relevant. Zu einem angegebenen Wort lieferte der Webservice eine Liste mit GebärdenSchrift-Zeichen zurück. Der Webservice war dabei durch einen Service der Firma Canoo[5] in der Lage, die unflektierte Form des angegebenen Wortes zu finden. Auf diese Weise konnte zum Beispiel das Wort „gingen“ auf die Grundform „gehen“ abgebildet und anhand letzterer in der Datenbank nach einer Übersetzung gesucht werden.

Während der Arbeit an dem im Rahmen dieser Arbeit entwickelten GebärdenSchrift-Editor haben sich jedoch die Anforderungen an den Webservice verändert bzw. sie wurden erweitert. Die Anpassung des Webservices war im Rahmen dieser Arbeit nicht möglich. Dieser Abschnitt beschreibt die erweiterten Anforderungen.

Die wichtigste neue Anforderung betrifft die Speicherung der Gebärden. Bisher war vorgesehen, die Gebärden als Bilddateien im SVG-Format abzuspeichern. Das vektorbasierte SVG-Format hat den Vorteil, dass es sich ohne Qualitätsverlust beliebig skalieren lässt. Bei diesem Format werden nicht mehr die Pixel und ihre Farben gespeichert, sondern durch Parameter beschriebene Formen und Linien. Diese so genannten Pfade werden erst bei der Anzeige abhängig von der Zoom-Stufe auf die Pixel des Anzeigemonitors umgerechnet. Das auf XML basierende SVG-Format sollte zusätzlich mit Informationen angereichert werden, welche die spätere Rekonstruktion der Struktur der Gebärde also die Zerlegung in die einzelnen Symbole ermöglichen sollte.

Während der Entwicklung des GebärdenSchrift-Editors hat es sich jedoch als sinnvoll erwiesen, die Struktur der Gebärde und ihre Darstellung zu trennen. Wie in Abschnitt 5.3.1 beschrieben wird in der von mir entwickelten Anwendung eine Gebärde auf logischer Ebene als eine Verbindung von GebärdenSchrift-Symbolen mit bestimmten Koordinaten beschrieben. Bei der Anzeige wird dann nicht die Gebärde als ganzes, sondern die einzelnen Symbole an der entsprechenden Position angezeigt. Dies hat folgende Vorteile: Einen eigenen vektorbasierten Zeichensatz für die GebärdenSchrift-Symbole zu erstellen ist aufgrund ihrer großen Anzahl sehr aufwändig. Die Trennung von Struktur und Darstellung hat den Vorteil, dass für die Darstellung der einzelnen Symbole auf die in Abschnitt 2.3.2 beschriebene Symboldatenbank zurückgegriffen werden kann, bis ein Zeichensatz in besserer Qualität vorliegt. Die Speicherung von Gebärden ist zudem unabhängig von dem Bild-Dateiformat, in dem die Symbole vorliegen. So wären auch verschieden gestaltete Zeichensätze möglich, vergleichbar mit verschiedenen Schrifttypen in der lateinischen Schrift. Des Weiteren bietet das in Abschnitt 3.2.3 beschriebene SignPuddle Online die

[5] `http://canoo.com/wmtrans/home/index.html`

Möglichkeit, deren Gebärdendatenbank in einem XML-Format zu exportieren, das nur die Struktur enthält und in den Webservice importiert werden könnte.

Mit der Trennung von Struktur und Darstellung der Gebärden wird es auch erforderlich, dass der Webservice nicht nur die Gebärden, sondern auch die einzelnen Symbole liefert. Zwar stammen die Bilder für diese momentan aus der ISWA-Datenbank, jedoch kommen die Identifikationscodes der Symbole, die für den Zugriff auf die Datenbank benötigt werden, vom weiter unten beschriebenen Dummy-Servlet, das momentan den Webservice vertritt. Lieferte der Webservice diese Identifikationscodes, wäre es möglich, die Symbole individuell zu sortieren und zu filtern. Denkbar wäre eine Filterung nach DGS-relevanten Symbolen oder eine Sortierung nach Verwendungshäufigkeit.

Eine weitere neue Anforderung ergibt sich aus der in den vorherigen Abschnitten beschriebenen, sinnvollen, getrennten Behandlung von Wörtern und Sonderzeichen. Ein Sonderzeichen wird bei der Übersetzung nur auf ein Symbol der GebärdenSchrift abgebildet, ein Wort dagegen auf eine oder mehrere Gebärden, die jeweils aus mehreren Symbolen bestehen. Bisher liefert der Webservice nur verschriftlichte Gebärden zu entsprechenden Wörtern, eine spezielle Behandlung von Sonderzeichen fehlt.

Des Weiteren enthielt der Webservice einige Einschränkungen, die sich als nicht sinnvoll erwiesen haben. Um doppelte Datenbankeinträge zu vermeiden, ließ sich die gleiche Gebärde nur einmal in der Datenbank speichern. Zudem hatte jede Gebärde nur eine festgelegte lautsprachliche Bedeutung. In Abschnitt 2.1.2 habe ich jedoch angedeutet, dass ein und dieselbe Gebärde für ein Verb und ein Nomen stehen kann. Dies kommt in der DGS sehr häufig vor. Nach den zuvor beschriebenen Einschränkungen, wäre es jedoch nicht möglich, die gleiche Gebärde mit der Bedeutung „arbeiten“ und der Bedeutung „Arbeit“ in der Datenbank zu führen. Aus diesem Grund ist es notwendig, dass eine Gebärde mehrere Bedeutungen haben kann.

Weiterhin erlaubte der Webservice bisher nur das Abspeichern der Gebärden. Darüber hinaus wäre es jedoch sinnvoll, auch die erstellten Texte bzw. Übersetzungen auf dem Webservice abszuspeichern und zu verwalten.

Zusammenfassung der neuen Anforderungen an den Webservice:

- Trennung von Struktur und Darstellung bei der Abspeicherung von Gebärden
- Bereitstellung der einzelnen GebärdenSchrift-Symbole
- Behandlung von Sonderzeichen
- Angabe mehrerer lautsprachlicher Bedeutungen für eine Gebärde
- Speicherung der GebärdenSchrift-Texte

Der Webservice wird momentan noch zu Testzwecken durch ein Dummy-Servlet ersetzt. Dieses stellt dennoch schon viele erforderlichen Daten bereit. Es verwendet die zuvor erwähnte Gebärdendatenbank des in Abschnitt 3.2.3 beschriebenen SignPuddle Online und kennt damit schon in etwa 8.000 Gebärden. (Aufgrund von Kodierungsfehlern im SignPuddle-Exportformat konnten vorerst nicht alle 11.000 Gebärden importiert werden.)

Die Gebärden werden aus einer Datei ausgelesen und bei entsprechenden Wortanfragen zurückgeliefert. Eine Flektionsauflösung der eingegebenen Wörter findet dabei noch nicht statt. Des Weiteren liefert das Dummy-Servlet wie zuvor erwähnt eine Auswahl von GebärdenSchrift-Symbolen, die im `KeyboardPanel` dargestellt werden und so in Gebärden eingefügt werden können. Dazu werden zu Testzwecken systematisch in etwa 400 Symbolcodes erzeugt. Dies entspricht ungefähr zwei Dritteln der 639 Basissymbole des ISWA. Eine Filterung nach DGS-relevanten Symbolen findet nicht statt.

6 Fazit und Ausblick

Diese Arbeit hat zunächst ausführlich den fachlichen Hintergrund und den Anwendungskontext beschrieben. Anschließend wurden daraus resultierende Anforderungen für die IT-basierte Anfertigung von GebärdenSchrift-Texten abgeleitet und aktuelle Software vor diesem Hintergrund bewertet. Anhand der so gewonnenen Erkenntnisse habe ich einen GebärdenSchrift-Editor entworfen und prototypisch umgesetzt.

Neben der aktuellen GebärdenSchrift-Software bewerte ich auch den entwickelten Prototypen anhand der zuvor definierten Anforderungen. Abbildung 27 fasst die Ergebnisse zusammen.

		SignWriter DOS	SignWriter Java	SignPuddle Online	Eigener Entwurf
1.	Abspeichern und Einfügen vordefinierter verschriftlichter Gebärden	+	+	++	+
2.	Unterstützung bei der Übersetzung in LBG/DGS	-	-	-	++
3.	Erstellen und Editieren handschriftlicher Gebärden	+	+	++	++
4.	Editieren der Gebärden während der Texterstellung	++	++	- -	++
5.	Komfortabler Zugriff auf die GebärdenSchrift-Symbole	+	-	+	+
6.	Anschauliche und selbsterklärende Steuerung	- -	-	+	++
7.	Effiziente Steuerung	++	-	- -	++

Abbildung 27: Zusammenfassende Bewertung des eigenen Entwurfs für einen GebärdenSchrift-Editor.

Dank des selbstentwickelten Tastaturfeldes wurde eine anschauliche und effiziente Steuerung erreicht. Weiterhin war die Unterstützung der Übersetzung lautsprachlicher Texte in LBG und DGS, repräsentiert in der GebärdenSchrift, eine wichtige Problemstellung. Diese konnte ebenfalls aufgrund der direkten Übersetzung von eingegebenem Text sehr gut gelöst werden. Obwohl auch die restlichen Anforderungen weitestgehend erfüllt sind, gebe ich im Folgenden einen Ausblick über mögliche Weiterentwicklungen.

Die Steuerung des GebärdenSchrift-Editors erfolgt bisher nahezu ausschließlich über die Tastatur. Trotz der anschaulichen Präsentation durch das Tastaturfeld würde eine zusätzliche Maussteuerung die Einsteigerfreundlichkeit des Editors weiter verbessern. Denn die direkte Manipulation beim Verschieben von Symbolen nach dem Prinzip Drag & Drop sind viele Benutzer aus Grafikanwendungen oder von Desktopsystemen gewohnt. Darüber hinaus würde eine Möglichkeit zur gleichzeitigen Selektion und Manipulation mehrerer GebärdenSchrift-Symbole das Editieren von verschriftlichten Gebärden weiter erleichtern. Dies könnte zum Beispiel wie von anderen Programmen gewohnt mit der zuvor erwähnten Maussteuerung durch Aufziehen eines Selektionsrahmens realisiert werden.

Wie in Abschnitt 5.3.3 beschrieben, steht die Entwicklung eines an die dort aufgeführten Anforderungen angepassten Webservices noch aus. Damit einher geht die Abspeicherung und Verwaltung sowohl der GebärdenSchrift-Texte als auch der einzelnen verschriftlichten Gebärden. Ebenso wäre dort eine sinnvolle Filterung und Sortierung der GebärdenSchrift-Symbole möglich. Die Voraussetzungen für die Anbindung eines solchen Webservices wurden im entwickelten Prototypen bereits geschaffen.

Ebenso sind das Ausdrucken der erstellten GebärdenSchrift-Texte oder das Exportieren zum Beispiel im PDF-Format wichtige Voraussetzungen für die Erstellung von Unterrichtsmaterialien. Im entwickelten Prototypen stand jedoch vor allem die Bedienung des GebärdenSchrift-Editors im Vordergrund. Die zuvor beschriebenen Speicherungs- und Ausgabefunktionen sind Standardfunktionen. Deshalb wurden sie bei der Planung weniger priorisiert und nicht realisiert.

Die Texterstellung und speziell die Übersetzung sind mit dem Prototypen bereits jetzt sehr viel effizienter durchführbar als mit existierender Software. Im Folgenden kann der Prototyp genutzt werden, um das verwendete Konzept der Steuerung zu erproben und weiterzuentwickeln.

Abbildungsverzeichnis

1 Die Gebärde für „besitzen“ wird durch die zu einem „Sch“-Laut passende Mundgestik begleitet. (Quelle: BKPZ06, S. 98) 9

2 Die Gebärde für „schwimmen“. (Quelle: BKPZ07, S. 148) 11

3 Die alphaförmige Bewegung bei der Inkorporation der Verneinung: Die Gebärden für „stimmen“ (links) und „nicht stimmen“ (rechts). (Quelle: BKPZ07, S. 44/130) 11

4 Drei Beispiele für Transferverben: die Gebärden für „schicken“, „geben“ und „zeigen“ (von links nach rechts). Die verschiedenen Pfeile beim rechten Beispiel deuten an, wie die Bewegungsrichtung verändert werden kann. (Quelle: BKPZ07, S. 69) 12

5 Zwei Gebärden mit der Bedeutung „Bank“: links die Sitzgelegenheit, rechts das Geldinstitut. (Quelle: BKPZ07, S. 43/120) 14

6 Die Gebärden für „bitte“ repräsentiert in der GebärdenSchrift (rechts). (Quelle: BKPZ06, S. 38) 15

7 Die drei Grundformen der Hand (links) und jeweils eine Variation (rechts). (Quelle: Wöh05) 16

8 Beispiele für verschiedene Handstellungen. (Quelle: vgl. Wöh05, S. 54,55) 17

9 Zwei GebärdenSchrift-Symbole und ihr Identifikationscode im ISWA. . . 20

10 Die Gebärde für „geben“ in der Variante „P-ich-GEBEN-P-du“. (Quelle: BKPZ07, S. 69) 22

11 Die Gebärde für „mühsam“ notiert in HamNoSys (rechts). (Quelle Zeichnung: BKPZ06, S. 69)(Quelle HamNoSys-Schriftzug: fDGuKG04) 22

12 Ein Diktat in GebärdenSchrift. (Quelle: Stefan Wöhrmann) 25

13 Die leere Schreibfläche von SignWriter DOS. 30

14 Tippen in GebärdenSchrift mit SignWriter DOS. 31

15 SignWriter Java unter Windows XP: Menüpunkte können mit Hilfe GebärdenSchrift benannt werden. 33

16 SignWriter Java unter Windows XP: Das Erstellen und Editieren von Gebärden erfolgt in einer Editierbox über der Schreibfläche (cyanfarben). . . 34

17 Der SignMaker von SignPuddle Online (Browser: Mozilla Firefox). . . . 35

18 Der modale Dialog zum Auswählen der alternativen Übersetzungen bei der Übersetzung mit SignPuddle Online. 36

19 Zusammenfassende Bewertung der existierenden GebärdenSchrift-Software. 37

20 Der Übersetzungsmodus: Der eingetippte Text (oben) wird direkt in GebärdenSchrift übersetzt (unten). 39

21 Der Bearbeitungsmodus mit dem Tastaturfeld (oben). Das orange gefärbte Symbol (unten) ist selektiert und kann manipuliert werden. 40
22 Die Funktionsweise des Tastaturfeldes: Der Benutzer drückt zunächst die Taste „1“ und betritt die Kategorie „Zeigefinger“. Anschließend fügt er das zweite Symbol mit der Taste „2“ ein. 41
23 Die Architektur des GebärdenSchrift-Editors lässt sich in die Komponenten Werkzeug, Material und Webservice unterteilen. 45
24 Die Material-Komponente SignWiritingDocument. 46
25 Die Material-Komponente Sign. 47
26 Die zentralen Klassen innerhalb der Werkzeug-Komponente (zusätzlich angedeutet: Referenzen auf die Materialien und den Webservice). 48
27 Zusammenfassende Bewertung des eigenen Entwurfs für einen Gebärden-Schrift-Editor. 52

Literatur

[BKPZ06] Anne Beecken, Jörg Keller, Siegmund Prillwitz und Heiko Zienert. *Grundkurs Deutsche Gebärdensprache, Arbeitsbuch Stufe I*. Band 3 in Gebärdensprachlehre. Signum Verlag, 2. Auflage, 2006.

[BKPZ07] Anne Beecken, Jörg Keller, Siegmund Prillwitz und Heiko Zienert. *Grundkurs Deutsche Gebärdensprache, Arbeitsbuch Stufe II*. Band 5 in Gebärdensprachlehre. Signum Verlag, 2007.

[BS07] Christian Borgwardt und Simone Schulz. Die Schulbildung Gehörloser in der BRD. *Grundkurs Deutsche Gebärdensprache, Arbeitsbuch Stufe II*, Seiten 78–82, 2007.

[fDGuKG04] Universität Hamburg: Institut für Deutsche Gebärdensprache und Kommunikation Gehörloser. *Notation von Gebärdensprache: HamNoSys*. `http://www.sign-lang.uni-hamburg.de/projekte/hamnosys/`, Januar 2004. Zuletzt besucht am 21.12.2009.

[fJ02] Bundesministerium für Justiz. *Gesetz zur Gleichstellung behinderter Menschen*. `http://www.gesetze-im-internet.de/bgg/BJNR146800002.html`, 2002. Zuletzt besucht am 23.12.2009.

[Fri07] Daniel Frick. Haarschneiden oder Wurst essen? `http://www.tagesschau.de/inland/meldung494482.html`, September 2007. Zuletzt besucht am 23.12.2009.

[Gup08] Vipul Gupta. *Accelerated GWT*. Apress, 2008.

[PvMMH08] Chrissostomos Papaspyrou, Alexander von Meyenn, Michaela Matthaei und Bettina Herrmann. *Grammatik der Deutschen Gebärdensprache*. Band 6 in Gebärdensprachlehre. Signum Verlag, 2008.

[Sle09] Stephen Slevinski. *SignWriting Image Server*. `http://www.signbank.org/swis/`, 2009. Zuletzt besucht am 23.12.2009.

[Wöh05] Stefan Wöhrmann. *Handbuch zur GebärdenSchrift*. Verlag Birgit Jacobsen, 2005.

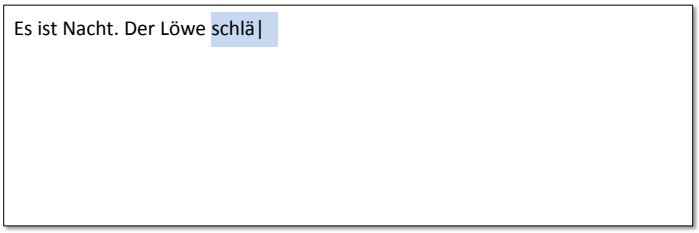
[Z+98] Heinz Züllighoven et al. *Das objektorientierte Konstruktionshandbuch*. Dpunkt Verlag, 1998.

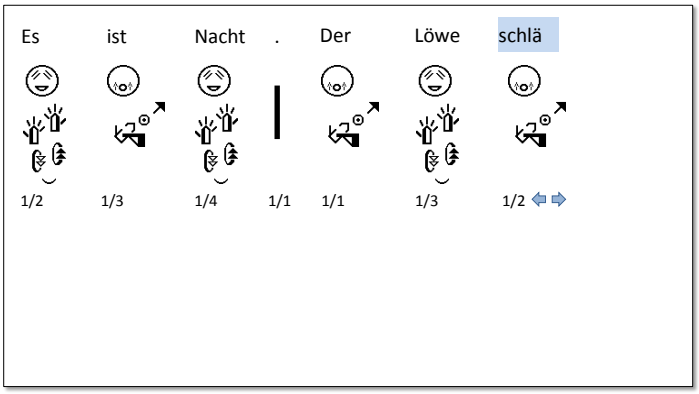
A Planung

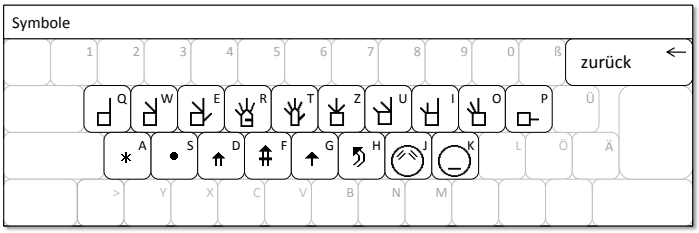
Story	Task	Aufwand (1=niedrig, 5=hoch)		Begründung für Priorisierung
Der Übersetzungsmodus (GebärdenSchrift schnell anhand der lautsprachlichen Bedeutung tippen)		15	1. Iteration	Stellt die Funktionalität des alten Prototypen dar. Ohne die Funtkionalität enthält der Prototyp keinen Mehrwert. Darüber hinaus ist diese Story Voraussetzung für weitere Stories und bildet die Story einen Querschnitt durch die Architektur.
	Aufbau der Oberfläche	3		
	Erstellung der Tokenbox (für strukturerhaltende Darstellung der Gebärde)	1		
	Nach Eingabe eines Wortes erscheint die entsprechende Tokenbox	1		
	Gleiches für mehrere Wörter	2		
	Tokenboxes können selektiert werden (entsprechend der Cursorposition im Editorfeld)	3		
	Erstellung von Dummy-Daten	3		
	Das Wort im Editorfeld und in der selektierten Tokenbox sind mit der gleichen Farbe hinterlegt	2		
	Alternative Übersetzungen lassen sich mit der Tastatur durchschalten	3		
Wechsel in den Bearbeitungsmodus		8		
	Wechseln in den Bearbeitungsmodus durch Drücken einer Taste: Das (vorerst leere) Tastaturfeld erscheint	2		
	Den Bearbeitungsmodus per Esc-Taste verlassen	1		
	Je nach Modus unterschiedliche Behandlung der Tastaturevents	4		
	Farbliche Hinterlegung der Gebärde im Bearbeitungsmodus	1		
Erstellen neuer Gebärden bzw. Einfügen/Tippen von GebärdenSchrift-Symbolen mit Hilfe des Tastaturfeldes		16	2. Iteration	Beinhaltet das zentrale Feature des Prototypen. Risikominimierung durch frühes exploratives Testen der Steuerung über das Tastaturfeld.
	Bei unbekannten Wörtern erscheint eine leere Gebärde	1		
	Drücken einer (nicht definierten) Taste im Bearbeitungsmodus fügt ein Symbol zur selektierten Gebärde hinzu	3		
	Tastaturfeld gleicht optisch einer Tastatur	5		
	Eine Auswahl von Symbolen auf die Tasten des Tastaturfeldes verteilen	3		
	Navigation im Symbolbaum, Öffnen von Kategorien, entsprechende Aktualisierung des Tastaturfeldes	4		
GebärdenSchrift-Symbole verschieben und anpassen		18	3. Iteration	Funktionen, die zum Erstellen neuer Gebärden entscheidend sind, jedoch im Prinzip ähnlich funktionieren wie in bisherigen Programmen. Daher kein sehr großes Alleinstellungsmerkmal. Werden jedoch zwingend zum Erstellen von Gebärden benötigt -
	Symbole (in der Tokenbox) können selektiert werden, eingefügte Symbole sind selektiert	4		
	Selektierte Symbole werden farblich hervorgehoben	2		
	Selektierte Symbole können mit den Pfeiltasten verschoben werden	3		
	Mit Hilfe der Tab-Taste lassen sich die Symbole einer Gebärde nacheinander auswählen	4		
	Selektierte Handsymbole können mit Hilfe der Tastatur entsprechend der verschiedenen Handstellungen angepasst werden	3		

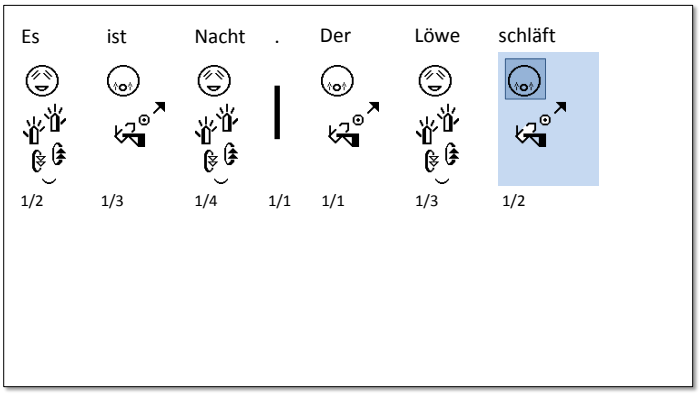
Selektierte Symbole können mit der Entf-Taste entfernt werden	2		daher vorgezogen.
Optimierung des Tastaturfeldes	15	4. Iteration	Funktionen, die die Bedienung selbsterklärender machen. Verfeinerung des Tastaturfeldes. (Später in eigene Story ausgelagert und nach hinten verschoben, daher teilweise bereits erledigt.)
Anzeige des Pfades in der Navigationsleiste	2		
Tasten für Kategorien unterscheiden sich optisch von Tasten für Symbole	1		
Nicht belegte Tasten werden im Tastaturfeld ausgegraut	2		
Für bekannte Wörter lässt sich per Tastendruck eine neue alternative Gebärde anlegen	3		
Tastaturfeld visualisiert alle Tasten	3		
Bessere Dummy-Daten	4		
Speichern von Texten und neuen bzw. geänderten Gebärden in der Datenbank	16	5. Iteration	Im fertigen Programm eine Selbstverständlichkeit. In einem Prototypen zur Demonstration der Programmfunktionalität (Bedienung über das Tastaturfeld, Erstellen von GebärdenSchrift-Texten) jedoch nicht entscheidend. Daher geringere Priorisierung.
Strg+S öffnet im bearbeitungsmodus einen den Dialog zum Speichern der Gebärde	2		
Dialog zum Speichern von Gebärden (Sinnvolle Angaben überlegen)	5		
Service zum Speichern der Gebärde	3		
Strg+S öffnet im Übersetzungsmodus einen Dialog zum Speichern des Textes	2		
Auswahl der alternativen Übersetzungen bleibt bei Speichern erhalten	4		

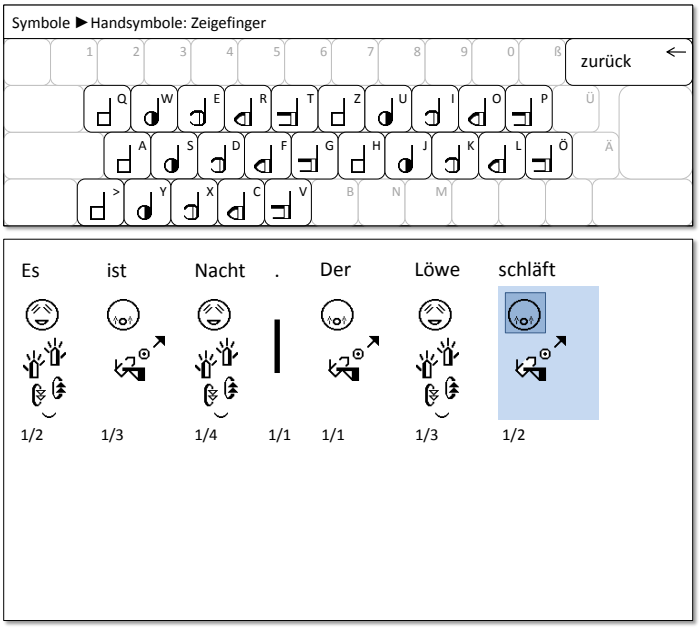
B Mock-Up

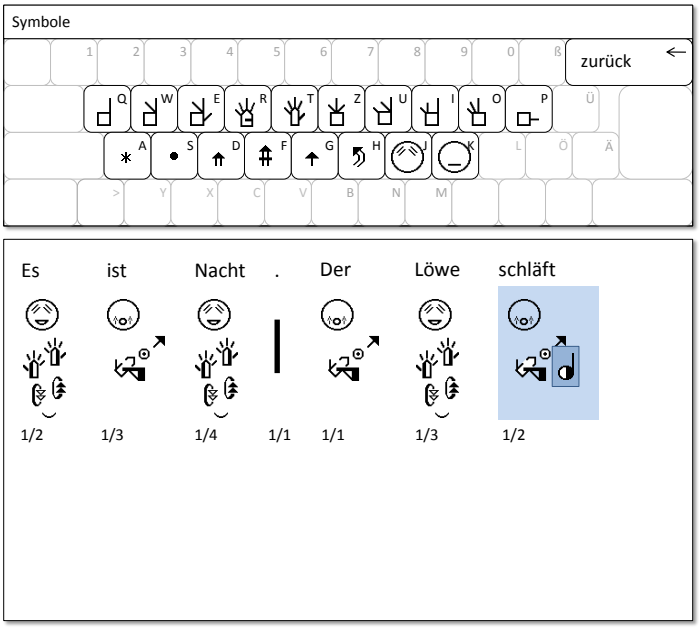

Es ist Nacht. Der Löwe schlä|


Es
ist
Nacht
.
Der
Löwe
schlä
1/2
1/3
1/4
1/1
1/1
1/3
1/2


Symbole
zurück


Es
ist
Nacht
.
Der
Löwe
schläft
1/2
1/3
1/4
1/1
1/1
1/3
1/2


Symbole ▶ Handsymbole: Zeigefinger
zurück
Es
ist
Nacht
.
Der
Löwe
schläft
1/2
1/3
1/4
1/1
1/1
1/3
1/2


Symbole
zurück
Es
ist
Nacht
.
Der
Löwe
schläft
1/2
1/3
1/4
1/1
1/1
1/3
1/2

Erklärung

Ich versichere, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Hamburg, den 23. Dezember 2009