

PIA - Eine transparente Benutzungsschnittstelle für motorisch behinderte Personen

Christian Piwetz

Universität - Gesamthochschule - Essen

Zusammenfassung

Moderne graphische Benutzungsoberflächen und Anwendungsprogramme sind meist nicht darauf ausgelegt, von Menschen mit Behinderungen bedient zu werden. In diesem Beitrag wird eine prototypische Eingabehilfe für den Apple Macintosh vorgestellt, die motorisch behinderten Menschen die Texteingabe erleichtern soll. Der Anwender interagiert nicht mehr direkt mit dem Anwendungsprogramm, sondern mit Hilfe der Benutzungsschnittstelle, die als eigenständige Schicht über dem Anwendungsprogramm liegt. Die Verbindung zwischen dem Anwendungsprogramm und der Eingabehilfe erfolgt über einen Standardmechanismus des Macintosh-Betriebssystems, so daß prinzipiell alle Programme mit der Eingabehilfe bedient werden können. Zusätzlich werden kontextabhängige Vorhersagen generiert, die der Anwender sehr leicht in seinen eigenen Text übernehmen kann.

1 Einleitung

Die in den letzten Jahren auf den Markt gebrachten Computer, Benutzungsoberflächen und Anwendungsprogramme rühmen sich damit, besonders anwenderfreundlich zu sein. Dies trifft sicherlich zu, wenn ein Anwender dieser Computer und Programme beide Arme vollständig kontrollieren kann. Ist der Anwender jedoch vom Hals abwärts gelähmt, kann er diese Computer geschweige denn eines dieser „modernen“ Anwendungsprogramme nicht bedienen. Wie Glinert und York in [8] schreiben: „Most hardware and software has been, and continues to be, designed by and for people who are not disabled.“

Gerade aber der Computer stellt die wichtigste technische Entwicklung der letzten Jahre dar, die in vielen Bereichen des täglichen Lebens zum Einsatz kommt. Durch die fehlende Unterstützung von Menschen mit Behinderungen wird eine große Bevölkerungsgruppe von den Möglichkeiten des Computers ausgeschlossen.

Das hier vorgestellte Programm PIA („Predictive Interface Agent“) stellt eine Benutzungsoberfläche für motorisch behinderte Personen dar, die die Standardtastatur nur langsam, mit Mühe oder gar nicht bedienen können. Ausgangspunkt für die Entwicklung von PIA ist das Programm *Reactive Keyboard* [6, 7], das an der Universität Calgary entwickelt wurde. Während beim *Reactive Keyboard* die Textvorhersage im Vordergrund stand, lag der Schwerpunkt bei der Entwicklung von PIA bei der generellen Unterstützung der Texteingabe für verschiedene

Behinderungen und der vollständigen Loslösung der Eingabehilfe von dem jeweiligen Anwendungsprogramm (Textverarbeitung, Tabellenkalkulation usw.). „Transparent“ bedeutet also in diesem Zusammenhang, daß das eigentliche Anwendungsprogramm nichts von der Benutzungsschnittstelle PIA bemerkt.

Die beschriebene Software wurde während einer Kooperation der Universitäten Dortmund und Essen mit dem Forschungsinstitut Technologie-Behindertenhilfe, Wetter-Volmarstein entwickelt und war Teil der Diplomarbeit des Autors [17].

Im folgenden zweiten Abschnitt werden neben einer Erläuterung der möglichen Eingabestrategien bereits existierende Eingabehilfen für Menschen mit Körperbehinderungen besprochen. Der dritte Abschnitt beschreibt die funktionalen Anforderungen an PIA. Die daraus resultierenden Oberflächenelemente werden im vierten Abschnitt erläutert. Im fünften Abschnitt werden einige Aspekte der Implementierung beschrieben. Die Schlußbetrachtung gibt einen Ausblick auf die mögliche weitere Entwicklung von PIA.

2 Texteingabehilfen für körperbehinderte Menschen

In diesem Kapitel werden zuerst die verschiedenen Eingabestrategien für Computerprogramme erläutert. Danach erfolgt ein kurzer Überblick über bereits existierende Eingabehilfen für Menschen mit Körperbehinderungen.

2.1 Eingabestrategien

Bei der Texteingabe mit dem Computer gibt der Anwender Buchstaben und Ziffern ein, die die einzelnen Worte und Sätze des Textes bilden. Zusätzlich macht er Eingaben, die nicht der Texterstellung dienen, d.h. er löscht Zeichen, bewegt die Schreibmarke oder formatiert den Text. Die Zahl der für die Texterstellung möglichen Zeichen beträgt für die deutsche Sprache mindestens 45: die 26 Buchstaben, das Leerzeichen, die Umlaute und das ß sowie die Satzzeichen. Hinzu kommen noch die zehn Ziffern. Wie soll nun ein Computerbenutzer mit Behinderungen, der keine Standardtastatur sondern nur wenige Schalter bedienen kann, diese verschiedenen Zeichen eingeben können? Hierbei werden drei Eingabestrategien [1] unterschieden:

Direkte Auswahl

Es steht für jedes Zeichen jeweils ein Schalter zur Verfügung. Beispiele sind die normale Tastatur, aber auch eine am Bildschirm angezeigte Tastatur, die über ein

Zeigegerät bedient wird. Großbuchstaben und Sonderzeichen werden über zusätzliche Umschalter erzeugt.

Codierung

Jedem Auswahlelement wird ein spezieller Code zugeordnet, den der Benutzer für die Auswahl eingeben muß. So ordnet das Morsealphabet jedem Zeichen eine Kombination aus langen („Strichen“) und kurzen („Punkten“) Impulsen zu.

Scanning

Die zur Auswahl stehenden Zeichen werden zeilen- und spaltenweise angeordnet. Eine Marke durchläuft in einem bestimmten Takt das Zeichenfeld zuerst zeilen- und dann spaltenweise. Durch Betätigung eines Tasters wählt der Anwender zuerst die Zeile und danach die Spalte aus, in der das gewünschte Zeichen steht.

2.2 Vorhandene Texteingabehilfen

Menschen mit Körperbehinderungen bilden im Gegensatz z.B. zu der Gruppe der blinden Menschen eine sehr heterogene Gruppe, was sich u.a. an der Vielzahl der verfügbaren Eingabehilfen widerspiegelt. Zur Verdeutlichung zwei Beispiele: für einen Computerbenutzer mit einem gelähmten Arm reicht schon Zusatzsoftware aus, die ihm die sequentielle Eingabe von Text ermöglicht. Dagegen ist der Hard- und Softwareaufwand für einen Menschen, der vom Hals abwärts gelähmt ist, sehr viel größer.

Verschiedene Hardwarehilfen (Folientastatur, Fußastatur, Links- oder Rechtshändertastatur etc.) erweitern bzw. ersetzen die Standardastatur. Sehr häufig kann aber die Eingabestrategie *direkte Auswahl*, die diesen Eingabegeräten zugrunde liegt, nicht mehr verwendet werden. Für die Kodierung oder das Scanning ist eine Kombination aus Hard- und Software notwendig. Die Auslösung der für das Scanning benötigten Takte ist auf unterschiedlichste Weisen möglich (Shein in [12]). Beispiele sind ein Schlagkissen (ein Impuls) [10], ein Saug-Blas-Rohr (je ein Impuls durch Saugen und Blasen) [23] oder ein durch die Zunge ausgelöster Schalter [5]. Die Umsetzung der Impulse muß über spezielle Software erfolgen.

Alle diese Eingabehilfen machen den Computer für einen behinderten Menschen erst nutzbar. Daneben existiert zur Beschleunigung der Texteingabe, die gerade mit der Eingabestrategie Scanning sehr mühselig und langsam ist, Zusatzsoftware, die Vorhersagen darüber erstellt, wie der gerade eingegebene Text weitergehen könnte. Der Bericht [13] der Universität Edinburgh enthält eine Übersicht über fast alle zum Erscheinungszeitpunkt erhältlichen Vorhersageprogramme für den

Apple Macintosh, MS-DOS-Computer und den in Großbritannien sehr verbreiteten BBC-Computer.

Alle diese Vorhersageprogramme können nur über die Tastatur oder die Maus bedient werden [13], unterstützen also nur die Eingabestrategie *direkte Auswahl*. Bei besonders schweren Behinderungen reicht dies jedoch nicht mehr aus.

3 Funktionelle Anforderungen an PIA

Im folgenden werden die Anforderungen an PIA erläutert, die für das Design der Benutzungsoberfläche und die Implementierung maßgeblich von Bedeutung waren. Dies beinhaltet neben den allgemeinen Anforderungen (transparenter Zugriff und Eingabeunterstützung durch Vorhersagen) die Definition von drei Benutzergruppen.

3.1 Transparenter Zugriff

Unter transparentem Zugriff (engl. transparent access [20, 22]) ist zu verstehen, daß die Eingabehilfe (PIA) für das eigentliche Anwendungsprogramm unsichtbar ist. PIA ist ein eigenständiges Programm, das zusätzlich zum Anwendungsprogramm auf dem Computer abläuft. Die Eingaben des Benutzers werden von PIA entgegengenommen und - zum Teil verändert - an die Zielanwendung weitergesendet. Diese verarbeitet die gesendeten Daten wie Eingaben von der Tastatur oder Maus. Die Zielanwendung „merkt“ also gar nicht, daß die Eingabe nicht direkt vom Benutzer, sondern von der Eingabehilfe PIA kommt. Zusätzlich kann der Anwender auch normal mit der Zielanwendung arbeiten. PIA bildet also eine zusätzliche Schicht zwischen der Anwendung und dem Benutzer (Abbildung 1, wobei Pfeile einen Datenfluß kennzeichnen).

Ein Interface Agent ist ein eigenständiges Programm, das für den Anwender selbständig Aufgaben ausführt [15]. PIA arbeitet dabei als Mittler zwischen Benutzer und Anwendungsprogramm [19]. Die Kommunikation zwischen PIA und der Zielanwendung sollte so realisiert werden, daß möglichst alle Anwendungsprogramme für den Apple Macintosh mit PIA zusammenarbeiten können. Dadurch ist der Anwender bei der Wahl der Zielanwendung keinen Beschränkungen unterworfen. Er kann z.B. mit der Textverarbeitung arbeiten, die seinen Ansprüchen am besten gerecht wird.

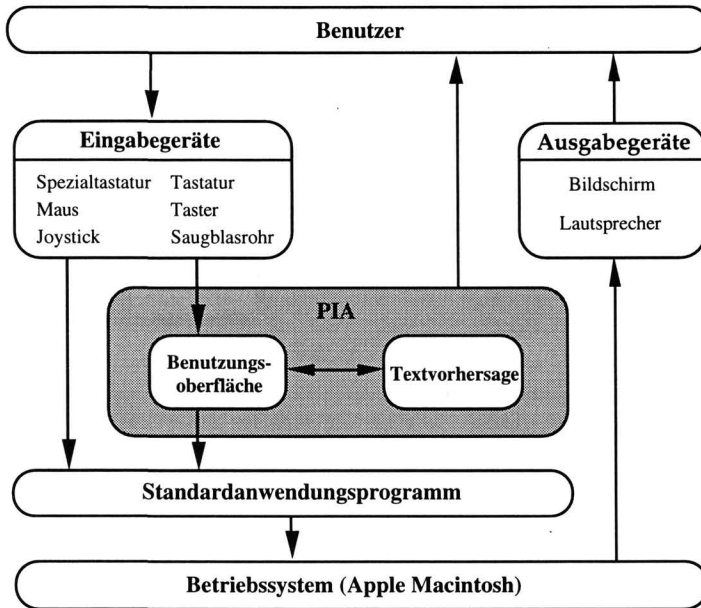


Abb. 1: Schematische Darstellung der Vorgehensweise bei PIA

3.2 Eingabeunterstützung durch Vorhersagen

Venkatagiri [22] hat gezeigt, daß durch zum Eingabekontext passende Vorhersagen die Anzahl der zu betätigenden Tasten / Schalter (Scanning) halbiert werden kann. Higginbotham [9] vergleicht fünf Vorhersageprogramme mit verschiedenen Vorhersagetechniken. Bei den dafür durchgeführten Tests wurden zwischen 36% und 41% der Tastenanschläge eingespart. Aufgrund dieser Ergebnisse ist es zweckmäßig, in PIA einen Vorhersagealgorithmus zu integrieren.

3.3 Die Benutzergruppen

Durch die Vielzahl der möglichen Behinderungen und Kombinationen von Behinderungen kann es kein allgemeines Benutzermodell für behinderte Computeranwender geben. Deshalb wurden für PIA die folgenden Benutzergruppen definiert:

- Benutzer von Zeigegeräten,
- Nur-Tastatur-Benutzer und

- Benutzer von Tastern.

Diese Einteilung stellt keine Klassifizierung von Menschen aufgrund körperlicher Merkmale dar, sondern dient nur zur Erstellung (grober) Benutzermodelle für die Implementierung.

Benutzer von Zeigegeräten

Der Anwender kann nur ein Zeigegerät (Maus, Trackball etc.) bedienen.

Nur-Tastatur-Benutzer

Anwender dieser Gruppe können zwar eine Tastatur benutzen, nicht aber ein Zeigegerät. Neben der Standardtastatur (evtl. mit einem Stirnstab) können hier aber auch Spezialtastaturen zum Einsatz kommen.

Benutzer von Tastern

Diese Gruppe kann nur eine bestimmte Anzahl Taster bedienen. Es ist unerheblich, um welche Art Taster es sich dabei handelt, z.B. ein Schlagkissen oder ein Saug-Blas-Rohr. In Abhängigkeit von der Anzahl der zu bedienenden Taster stehen dem Anwender verschiedene Eingabemöglichkeiten zur Verfügung.

Wenn der Benutzer mindestens fünf Taster bedienen kann, können vier davon für die Richtungssteuerung („Nach oben“, „Nach unten“, „Nach links“ und „Nach rechts“) verwendet werden. Der fünfte Taster dient der Bestätigung einer Aktion oder zur Signalisierung von Anfang und Ende einer Auswahl.

Ist der Anwender jedoch nur in der Lage, einen oder zwei Taster zu betätigen, muß die Bedienung der Benutzungsschnittstelle über die Eingabestrategie Scanning erfolgen. Beim automatischen Scanning mit einem Taster wird der Takt, mit dem die zur Auswahl stehenden Elemente durchlaufen werden, vom Computer vorgegeben. Desweiteren hat der Anwender die Möglichkeit, das Zeitintervall zwischen zwei Schritten durch die Betätigung eines zweiten Tasters selber festzulegen. Die Taktrate für den automatischen Durchlauf (z.B. 2 Sekunden) sollte jedoch an die Anforderungen des Anwenders angepaßt werden können.

Aufgrund der definierten Benutzergruppen ergaben sich zwei Eingabemodi für PIA. Zeigegeräte und die Tastatur stellen die Standardeingabegeräte für den Apple Macintosh dar. Wenn diese verwendet werden, verhält sich PIA aus Benutzersicht wie jedes andere Anwendungsprogramm. Der zweite Eingabemodus realisiert das Scanning für die Anwender von Tastern.

4 Die Benutzungsoberfläche von PIA

In PIA werden drei Fenster verwendet, die frei auf dem Bildschirm positioniert werden können. Das Hauptfenster, das als einziges von den drei Fenstern nicht ausgeblendet werden kann, ist das Vorhersagefenster, in dem die Vorhersagen und der aktuelle Eingabekontext angezeigt werden. Daneben kann der Anwender noch wahlweise die Bildschirmtastatur und die sog. Fernbedienung aufrufen. In diesem Abschnitt werden die Fenster und deren Bedienmöglichkeiten für die im vorigen Kapitel definierten Benutzergruppen beschrieben.

4.1 Das Vorhersagefenster

Das Vorhersagefenster (Abb. 2) zeigt neben dem Eingabekontext die aktuellen Vorhersagen. In der Titelleiste des Fensters wird der Name der Datei ausgegeben, in dem das aktuelle Vorhersagemodell gespeichert wird. In diesem Beispiel ist dies die Datei „Phrasen“. Die Vorhersagen werden sortiert nach der Wahrscheinlichkeit ihres Auftretens in einem nicht proportionalen Zeichensatz angezeigt, da so die einzelnen Zeichen der Vorhersagezeilen jeweils untereinander stehen. Es können maximal neun Vorhersagen auf einer Bildschirmseite angezeigt werden. Der Rollbalken am rechten Rand dient zum Blättern, wenn mehr als neun Vorhersagen generiert werden.

In dem oberen Textfeld wird der aktuelle Eingabekontext ausgegeben. Das kleine Dreieck zeigt das Ende des Kontextes an. Das Kontextfeld ist ein reines Ausgabefeld, der Anwender kann den Inhalt also nicht verändern. Wird die Return Taste (auf der Standard- oder der Bildschirmtastatur) betätigt, wird der Kontext in das Vorhersagemodell übernommen und das Kontextfeld gelöscht. Würde jedes eingegebene Zeichen oder jede ausgewählte Vorhersage sofort in das Vorhersagemodell eingefügt werden, würden auch alle Tippfehler des Anwenders übernommen werden. So hat der Anwender die Gelegenheit, Tippfehler zu verbessern. Das Vorhersagemodell kann dadurch bessere Vorschläge generieren.

Links neben den einzelnen Vorhersagezeilen sind die Beschriftungen „F1“ bis „F8“ zu sehen. Diese dienen als Erleichterung für die Auswahl von Vorhersagen über die Funktionstasten (der Standard- oder der Bildschirmtastatur). Es wird das erste Wort der entsprechenden Zeile ausgewählt, z.B. mit F5 das erste Wort der fünften Zeile.

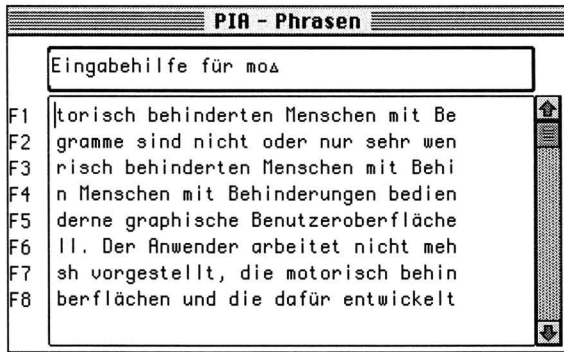


Abb. 2: Das Vorhersagefenster

4.2 Die Auswahl von Vorhersagen

Im folgenden werden für die einzelnen Gruppen die jeweiligen Auswahlmethoden beschrieben. Zu beachten ist dabei, daß sich die einzelnen Auswahlmethoden nicht immer eindeutig einer Benutzergruppe zuordnen lassen. So ist die Auswahl über die Funktionstasten primär für die Nur-Tastatur-Benutzer gedacht, Anwender eines Zeigegeätes oder des Scan-Modus können diese jedoch über die Bildschirmstastatur verwenden.

Benutzer von Zeigegeätes

Es werden zwar alle Begriffe für die Maus verwendet, wie z.B. „Mausknopf“, „Mauszeiger“ usw., gemeint sind damit jedoch alle Zeigegeätes. Um eine Vorhersage mit einem Zeigegeätes auszuwählen, bewegt der Anwender den Mauszeiger an den Anfang der entsprechenden Zeichenkette und drückt den Mausknopf. Durch Bewegung der Maus bei gleichzeitig festgehaltenem Mausknopf wird die Auswahl aufgezogen, d.h. der Text wird markiert. Wenn der Mausknopf gelöst wird, ist die Auswahl abgeschlossen, und der Text wird an die Zielanwendung gesendet. Durch einen Doppelklick auf die Beschriftungen „F1“ bis „F8“ am linken Rand der Vorhersagen wird das erste Wort der entsprechenden Zeile analog zum Verhalten der Funktionstasten ausgewählt.

Nur-Tastatur-Benutzer

Wie bereits erwähnt, ermöglichen die Funktionstasten *F1* bis *F8* die schnelle Auswahl des ersten Wortes einer Vorhersagezeile auf der aktuell angezeigten Seite. Eine individuelle Selektion innerhalb des Vorhersagefeldes kann der Anwender über die Pfeiltasten durchführen. Zur Markierung von Anfang und Ende

der Selektion ist eine weitere Taste notwendig. Diese Funktion wird bei der aktuellen Implementierung von der Taste *Einf.* (oberhalb der Pfeiltasten) übernommen.

Benutzer von Tastern

Kann der Anwender fünf Taster bedienen, erfolgt die Auswahl von Vorhersagen analog zu den Nur-Tastatur-Benutzern. Für Anwender von ein oder zwei Tastern werden die Vorhersagen von einer Marke zuerst zeilen- und dann spaltenweise durchlaufen. Zur Auswahl einer Zeile oder Spalte muß der Anwender den Auswahl-taster betätigen. Die aktuelle PIA-Implementierung arbeitet mit der Return-Taste als Auswahl- und mit der Leertaste als Takt-taster für den Scan-Modus mit einer benutzerdefinierten Taktrate. Sinnvoller ist hierbei jedoch die Unterstützung externer Taster.

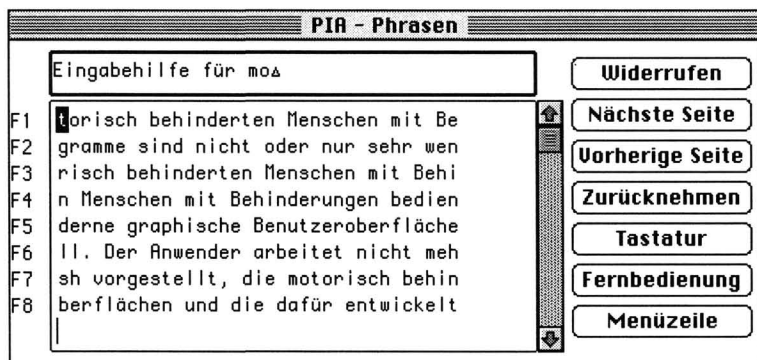


Abb. 3: Das Vorhersagefenster für den Scan-Modus

Das Vorhersagefenster wurde für den Scan-Modus rechts neben dem Rollbalken um sieben Knöpfe (Abb. 3) erweitert, die in den Scan-Durchlauf integriert sind. Diese Knöpfe ermöglichen das Widerrufen einer Auswahl, das Blättern innerhalb der Vorhersagen und den Wechsel zu den anderen PIA-Fenstern. Um Fehleingaben zu verhindern, haben alle anderen Tasten im Scan-Modus keine Funktion und werden ignoriert.

4.3 Die Bildschirmtastatur

Über die Bildschirmtastatur (Abb. 4) können die Benutzer von Zeigege-räten oder des Scan-Modus fast alle Zeichen und Tastenkombinationen der normalen Tastatur eingeben. Die Knöpfe werden entweder mit der Maus angeklickt oder im

Scan-Modus zuerst zeilen- und dann spaltenweise durchlaufen. Das Verhalten aller Knöpfe der Bildschirmtastatur entspricht dem der realen Tasten.

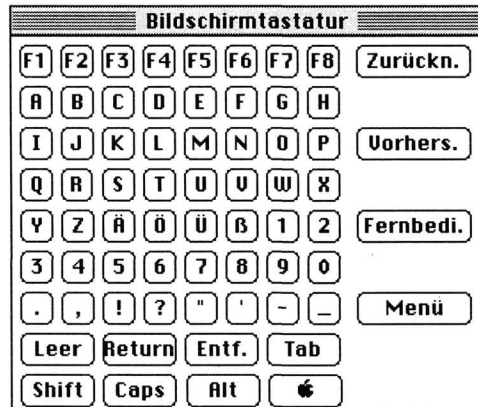


Abb. 4: Die Bildschirmtastatur von PIA

Die oberste Knopfleiste bilden die acht Funktionstasten. Ohne in den Vorhersagedialog zu wechseln, besteht so ein Zugriff auf die Vorhersagen. Die nächsten sieben Reihen bilden die Buchstaben, die Ziffern, die Satzzeichen und die Sondertasten „Leerzeichen“, „Return“, „Entfernen“ und „Tabulator“. Die Buchstabentasten sind bei der momentanen Implementierung alphabetisch angeordnet. Die letzte Knopfreihe beinhaltet die Sondertasten „Shift“ (Hochstelltaste), „Caps Lock“ (Feststelltaste), „Alt“ (Wahltaste) und die Apple-Taste (Befehlstaste). Das Verhalten dieser Tasten ist so implementiert, daß der Anwender den Text sequentiell eingeben kann, d.h. die Sondertaste, die der Anwender ansonsten selber festhalten muß, wird von PIA „festgehalten“. Die vier Knöpfe am rechten Rand der Tastatur haben dieselben Funktionen wie im Vorhersagedialog.

4.4 Die Fernbedienung

Die bisherigen Funktionen von PIA unterstützen nur die reine Texteingabe, nicht jedoch die Bearbeitung von Dateien (Anlegen, Öffnen, Speichern und Drucken). Ein Mausbenutzer kann mit einem Mausklick in die Zielanwendung wechseln und die gewünschte Funktion aufrufen. Ein Wechsel in die Zielanwendung ist jedoch gerade für Anwender des Scannings nicht möglich oder sehr problematisch, da nach dem Wechseln alle Eingabehilfen von PIA nicht mehr zur Verfügung stehen.



Abb. 5: Die Fernbedienung von PIA

Diese Funktionen der Zielanwendung können über das dritte PIA-Fenster - die sog. Fernbedienung (Abb. 5) - ausgeführt werden. Wie bei einer „echten“ Fernbedienung, z.B. für einen Fernseher, braucht sich der Anwender nicht zum eigentlichen Anwendungsprogramm (Gerät) zu begeben, sondern kann die gewünschte Funktion aus der „Ferne“ aufrufen. Die Fernbedienung ist wie das Vorhersagefenster und die Bildschirmtastatur in das Scanning eingebunden. Die einzelnen Knöpfe werden dabei von unten nach oben durchlaufen. Die Knöpfe können natürlich auch mit der Maus angeklickt werden. PIA wandelt die Benutzereingaben in entsprechende Anweisungen um und sendet diese an die Zielanwendung.

Während die Funktionen *Neu*, *Drucken* und *Beenden* ohne weitere Abfragen von der Zielanwendung ausgeführt werden können, benötigen die drei anderen Funktionen der Fernbedienung jeweils einen Dateinamen. Die Standarddialoge des Apple Macintosh für die Auswahl von Dateinamen können wegen der starken Fixierung auf die Mausbedienung nicht verwendet werden. Statt dessen wird oberhalb des Vorhersagefensters ein weiteres Dialogfenster eingeblendet. Nun kann der Anwender über die Standardtastatur, die Bildschirmtastatur oder durch die Auswahl von Vorhersagen den gewünschten Dateinamen eingeben. Durch eine erneute Betätigung des entsprechenden Knopfes in der Fernbedienung wird die Eingabe abgeschlossen und der Befehl an die Zielanwendung gesendet. Die übrigen Knöpfe der Fernbedienung haben dieselben Funktionen wie im Vorhersagedialog und in der Bildschirmtastatur.

5 Die Implementierung

In diesem Kapitel werden die Aspekte der Implementierung beschrieben, die für die Umsetzung der Anforderungen aus Kapitel 3 von besonderer Bedeutung sind.

5.1 Die Kommunikation mit der Zielanwendung

PIA sollte als eigenständige Anwendung mit möglichst allen Anwendungsprogrammen für den Apple Macintosh zusammenarbeiten, d.h. eine Anwendung (PIA) soll mit einer anderen Anwendung (der Zielanwendung) kommunizieren. Diese Art von Kommunikation wird unter dem Begriff *Inter-Application-Communication* (IAC) zusammengefaßt [2, 16].

Inter-Application-Communication

Umgangssprachlich bedeutet IAC nichts anderes, als daß zwei Anwendungsprogramme miteinander kommunizieren. Dabei ist es unerheblich, ob diese zwei Anwendungen auf demselben Rechner oder auf zwei durch ein Netzwerk miteinander verbundenen Rechnern laufen. Dies ist jedoch klar von Inter-Prozeß-Kommunikation (Inter-Process-Communication, IPC) und der Netzwerkschicht zu unterscheiden.

Während die Netzwerkschicht die physikalische Verbindung (z.B. Ethernet) und die Übertragungsprotokolle (z.B. TCP/IP) zur Verbindung der Computer behandelt, bezeichnet die Inter-Prozeß-Kommunikation die Sicht des Betriebssystems auf die einzelnen Prozesse und deren Behandlung. Dagegen werden beim IAC die Anwendungen nicht aus der Sicht des Programmierers („Wie sind die einzelnen Bits und Bytes angeordnet?“) betrachtet. Entscheidend ist hierbei, was sich in der Anwendung aus Benutzersicht befindet. Ein Anwendungsprogramm stellt ein sog. Kontextmodell (engl. context model) dar. Dieses besteht aus einzelnen Bestandteilen (Worte, Zeilen, Absätze usw.) und den Operationen, die darauf angewendet werden können, z.B. das Anlegen, Bewegen, Kopieren und Löschen von einzelnen Bestandteilen. IAC wird nun als die Kommunikation zwischen zwei Anwendungen mit verschiedenen Kontextmodellen definiert [16]. Im Gegensatz dazu arbeiten bei Netzwerkprotokollen beide Seiten der Kommunikation mit demselben Kontextmodell.

Apple Events und AppleScript

Das Betriebssystem des Apple Macintosh arbeitet ereignisgesteuert. Das Betriebssystem und alle aktiven Anwendungsprogramme befinden sich in Endlosschleifen und warten darauf, daß ein Ereignis (engl. event) eintritt. Ereignisse sind z.B. die

Betätigung einer Taste oder eine Bewegung der Maus. Das Betriebssystem gibt dieses Ereignis nach einer ersten Überprüfung an die betroffene Anwendung weiter, die das Ereignis dann bezogen auf den Anwendungskontext bearbeitet.

Eine spezielle Art von Ereignissen sind auf dem Apple Macintosh die sog. *Apple Events*, die zwischen zwei Anwendungsprogrammen gesendet werden können [2]. Apple hat diese Events als Standardweg für das IAC auf dem Apple Macintosh vorgesehen. Neben einer Datenübertragung können der Zielanwendung auch Anweisungen gegeben werden, z.B. „Drucke das Dokument mit dem Namen "Brief vom 1.12.1994" aus.“. Die Zielanwendung sollte solche Anweisungen genauso behandeln wie entsprechende Benutzereingaben über die Tastatur oder die Maus.

Der Mechanismus der Apple Events garantiert alleine noch keine allgemein gültige Anbindung von PIA an andere Anwendungsprogramme, da nicht davon ausgegangen werden kann, daß diese die benötigten Apple Events unterstützen. Auf Basis der Apple Events hat Apple die universelle Makrosprache AppleScript [3] entwickelt. AppleScript stellt für den Anwender eine Möglichkeit dar, einfache Programme für wiederkehrende Aufgaben (z.B. die Datensicherung) zu entwickeln und bei Bedarf auszuführen. Da AppleScript und somit Apple Events von Apple als Schlüsseltechnologien für Macintosh-Computer bezeichnet werden [18], kann man davon ausgehen, daß bald alle wichtigen Anwendungsprogramme AppleScript unterstützen und somit als Zielanwendungen für PIA in Frage kommen.

Die Verwendung von Apple Events für die Realisierung von PIA

Alle Datenübertragungen zwischen PIA und der Zielanwendung sind über entsprechende Apple Events realisiert worden. Dies schließt neben der Textübertragung von Tastatureingaben und Vorhersagen auch das Widerrufen von Eingaben sowie alle Funktionen der Fernbedienung zur Dateibehandlung ein. Eine tiefergehende Beschreibung findet man in [17].

5.2 Der Vorhersagealgorithmus

Grundlage und Motivation für die Entwicklung der Eingabehilfe PIA stellt das Programm *Reactive Keyboard* (RK) dar, beschrieben in [6] und [7]. Dieses wurde an der Universität Calgary, Kanada, von John J. Darragh und Ian H. Witten entwickelt. Die ersten Versionen des RKs für MS-DOS- und UNIX-Computer arbeiten als Eingabeunterstützungen für die Kommandozeileingabe. Neben den Kommandozeilenversionen des RKs wurde von Mark L. James eine Version für den Apple Macintosh mit dem Namen *RK-Pointer* entwickelt. Der unveränderte Vorhersagemechanismus ist in einen rudimentären Texteditor eingebettet. Die

Vorschläge werden in einem zusätzlichen Fenster angezeigt, während bei den anderen RK-Versionen nur jeweils eine Vorhersage ausgegeben wird.

Der Algorithmus des Reactive Keyboards

Der Vorhersagealgorithmus des RKs unterscheidet sich grundlegend von denen anderer Vorhersageprogramme (z. B. *Magic Typist* [4], *Co:Writer* [11] und *PAL* [14]), die fast alle auf der Basis von Wörterbüchern arbeiten. Bei diesen entnimmt der Algorithmus alle Wörter aus dem Wörterbuch, die zur aktuellen Eingabe des Benutzers passen. Wenn der Benutzer z.B. ein „h“ eingibt, sind dies alle Worte, die mit einem „h“ beginnen. Um die Vielzahl von möglichen Wörtern in eine sinnvolle Reihenfolge zu bekommen, ist jedem Wort eine Wahrscheinlichkeit zugeordnet. Nach dieser sortiert werden die einzelnen Worte angezeigt. Der Benutzer kann dann einen dieser Vorschläge annehmen oder aber ein weiteres Zeichen eingeben, z.B. ein „a“. Der Vorhersagealgorithmus sucht dann alle Worte mit dem Wortbeginn „ha“.

Das RK arbeitet hingegen nicht mit einem Wörterbuch, sondern sammelt alle Benutzereingaben in einem Vorhersagemodell. Dieses Vorhersagemodell wird durch einen Baum realisiert, in dem Zeichentupel der Länge k abgespeichert werden. Die sog. Modellgröße k gibt die Zeichenanzahl an, die zwischen dem aktuellen Eingabekontext und dem Inhalt des Vorhersagemodells verglichen werden. Wenn die Modellgröße z.B. gleich zehn ist, werden die letzten zehn Zeichen der Eingabe zur Überprüfung mit dem Vorhersagemodell herangezogen.

Das RK nimmt zur Erstellung von Vorhersagen einen reinen Zeichenvergleich vor. So kann der Fall eintreten, daß dadurch unsinnige Worte entstehen, die zwar auf Grund ihrer Zeichenfolge (syntaktisch) sehr gut in den aktuellen Kontext passen, aber keinerlei Bedeutung (semantisch) haben. Bei Vorhersageprogrammen auf Wörterbuchbasis kann dieser Fall nicht eintreten, da das Wörterbuch nur gültige Worte enthält. Auf der anderen Seite erkennt der Algorithmus des RK aufgrund des reinen Zeichenvergleichs nicht das Ende eines Wortes. So können nicht nur einzelne Worte, sondern ganze Phrasen und Sätze als Vorhersagen erstellt werden.

Das RK erstellt in einem Durchlauf 128 verschiedene Vorschläge, für jedes Zeichen der ASCII-Tabelle einen. Diese Vorschläge haben alle unterschiedliche Anfangszeichen, d.h. für ein mögliches Zeichen, das der aktuellen Eingabezeile folgen könnte, wird immer nur genau ein Vorschlag erstellt. Wenn der Benutzer beispielsweise ein „R“ eingibt, kann der erste Vorschlag des RKs „eactive Keyboard“ lauten. Es wird nun kein weiterer Vorschlag mit einem „e“ als ersten Buchstaben erzeugt, z.B. das Wort „Regenwald“. Das RK arbeitet adaptiv, d.h.

alle Vorhersagen werden aufgrund der vorangegangenen Benutzereingaben erstellt. Für eine genauere Beschreibung des Vorhersagealgorithmus wird auf die Originalliteratur [6, 7] verwiesen.

Gründe für die Wahl des RK-Algorithmus

Im Abschnitt 3.2 wurde bereits auf die Wichtigkeit von Textvorhersagen für die Eingabeunterstützung hingewiesen. Bei der Arbeit an PIA fiel die Wahl auf den RK-Algorithmus, da sich dieser durch seine adaptive Arbeitsweise von den anderen Vorhersagealgorithmen unterscheidet. Desweiteren ist der RK-Algorithmus im Quellcode verfügbar und konnte so leicht übernommen und angepaßt werden. Nur die Speicherverwaltung mußte wegen den geänderten Bedingungen unter Macintosh System 7 (32 Bit-Adressierung und Einbau eines Garbage Collectors) erweitert werden. Dadurch konnte der Schwerpunkt bei der Implementierung auf die Benutzungsschnittstelle, die Unterstützung verschiedener Eingabegeräte und die universelle Anbindung an Anwendungsprogramme gelegt werden.

Durch den Einsatz von Apple Events konnte eine sehr flexible und leistungsfähige Verbindung zwischen der Eingabehilfe PIA und der Zielanwendung realisiert werden. Die Unterstützung möglichst vieler Anwendungsprogramme für den Apple Macintosh kann als gegeben angesehen werden. Auch bieten die Apple Events noch viele Ansatzpunkte für Erweiterungsmöglichkeiten. Der Algorithmus des Reactive Keyboards stellt durch sein adaptives Vorhersagemodell eine interessante Alternative zu den anderen Vorhersagemethoden dar.

6 Schlußbetrachtung und Ausblick

PIA liegt als stabiler Prototyp in einer deutschen und einer englischen Version sowohl für den Apple Macintosh als auch für den Power Macintosh vor. Die Arbeiten haben die prinzipielle Realisierbarkeit einer separierten und vielseitig einsetzbaren Benutzungsschnittstelle für körperbehinderte Menschen nachgewiesen. Dennoch enthält PIA noch einige Teile, bei denen Verbesserungen notwendig sind.

Die Vorhersagen, die der Algorithmus des Reactive Keyboards erstellt hat, könnten durch einen zweiten Algorithmus ergänzt werden. Dieser fügt die gebräuchlichsten Wörter passend zum aktuellen Kontext in die Liste der Vorhersagen ein. Weitergehend könnten unsinnige Vorhersagen durch eine Rechtschreibprüfung bereits vor der Anzeige auf dem Bildschirm herausgefiltert werden. Das Layout der Bildschirmtastatur sollte über ein Zusatzprogramm an die Bedürfnisse und Wünsche des Anwenders angepaßt werden können. Die Menü-

zeile des Apple Macintosh kann nicht über die Tastatur (mit Ausnahme von Tastaturkürzeln) bedient werden. PIA sollte die Nur-Tastatur-Anwender und die Benutzer des Scan-Modus in die Lage versetzen, jeden Eintrag der Menüzeile zu erreichen. Daneben sollte auch die Redigierung von Text stärker berücksichtigt werden.

Literatur

- [1] N. Alm, J. L. Arnott, A. F. Newell. Predictional Conversational Momentum in an Augmentative Communication System. In *Communications of the ACM*, 35(5): 46-57, 1992.
- [2] Apple Computer, Inc. *Inside Macintosh: Interapplication Communication*. Addison-Wesley, 1993.
- [3] Apple Computer, Inc. *AppleScript-Getting Started Guide*. Cupertino, California, USA, 1993.
- [4] L. A. Bardi. *Magic Typist 2.0*. OLDUVAI Corporation, Miami, Florida, USA, 1990.
- [5] C. Clayton, R. G. S. Platts, M. Steinberg, A. M. Trudgeon. Palatal Tongue Controller. In *Computers for Handicapped Persons*. R. Oldenbourg, Wien, 1990.
- [6] J. J. Darragh, I. H. Witten. *The Reactive Keyboard*. Cambridge University Press, 1992.
- [7] J. J. Darragh, I. H. Witten, M. L. James. *The Reactive Keyboard*. In *IEEE Computer*, 23(11): 41-49, 1990.
- [8] E. P. Glinert, B. W. York. Computers and People with Disabilities. In *Communications of the ACM*, 35(5): 33-35, 1992.
- [9] D. J. Higginbotham. Evaluation of Keystroke Savings across Five Assistive Communication Technologies. In *AAC Augmentative and Alternative Communication of ISAAC*, 8(12):258-272, 1992.
- [10] Don Johnston Developmental Equipment, Inc. *Catalog 1993: Assistive Technology for Computer Access, Communication & Literacy*. Wauconda, Illinois, USA, 1993.
- [11] Don Johnston Developmental Equipment, Inc. *Co:Writer*. Wauconda, Illinois, USA, 1993.
- [12] M. Mantei, P. Orbeton. Human Factors in Computing Systems. Conference Proceedings of "Human Factors in Computing Systems", Zürich, Schweiz, 1986.
- [13] S. V. Millar, P. D. Nisbet. *Accelerated Writing for People with Disabilities*. CALL Centre, University of Edinburgh, 1993.
- [14] A. N. Newell, J. L. Arnott, L. Booth, W. Beattie, B. Brophy, I. W. Ricketts. Effect of the "PAL" Word prediction System on the Quality and Quantity of Text Generation. In *AAC Augmentative and Alternative Communication of ISAAC*, 8(12): 304-311, 1992.
- [15] J. Nielsen. Noncommand User Interfaces. In *Communications of the ACM*, 36(4): 83-99, 1993.

- [16] K. Piersol. Inter-Application-Communication. Videokassette. University Video Communication, Stanford, California, USA, 1992.
- [17] C. Piwetz. Entwicklung einer separierten Benutzungsschnittstelle für Textsysteme auf Basis des IAC-Mechanismus von Macintosh-System 7: Eine adaptive Eingabehilfe für motorisch behinderte Personen. Interner Bericht Nr. 7, Fachbereich Mathematik und Informatik / Systemmodellierung, Universität - Gesamthochschule - Essen, 1994.
- [18] R. Priem. Amber schmückt den Macintosh. In MACup, MACup Verlag, Hamburg, 7/1993: 12-13, 1993.
- [19] M. Streit. Architekturalternativen für Agentenbasierte Multimodale Interfaces. In J. Kunze, H. Stoyan (Hrsg.): KI-94 Workshop, Extended Abstracts. Springer-Verlag, 5-6, 1994.
- [20] U. Thakkar. Ethics in the Design of Human-Computer Interfaces for the Disabled. In SIGCAPH Newsletter, ACM Special Interest Group on Computers and the Physically Handicapped, 42: 1-7, 1990.
- [21] G. C. Vanderheiden. Making Software More Accessible for People with Disabilities. In SIGCAPH Newsletter, ACM Special Interest Group on Computers and the Physically Handicapped, 47, 1993.
- [22] H. S. Venkatagiri. Efficiency of Lexical Prediction as a Communication Acceleration Technique. In Augmentative and Alternative Communication, ISAAC, 9/1993:161-167, 1993.
- [23] VERTIKAL-Informatik. Kommunikationshilfen für Schwerbehinderte (Produktübersicht 1992). Weinsberg, Deutschland, 1992.

Christian Piwetz

Universität - Gesamthochschule - Essen

Fachbereich Mathematik und Informatik

45117 Essen

Tel.: 02 01 / 183 - 2168

e-mail: piwetz@informatik.uni-essen.de

