

Entwicklung und Evaluation eines alternativen Texteingabesystems für Persönliche Digitale Assistenten

Carsten Magerkurth

Fraunhofer IPSI, Darmstadt

Zusammenfassung

In diesem Beitrag wird die „MOBILE HOCHLEISTUNGS-TEXTEINGABE“ MOHOLT vorgestellt. MOHOLT ergänzt als tasten- und displaygesteuertes System die auf Persönlichen Digitalen Assistenten (PDAs) verbreiteten stiftbasierten Texteingabemethoden durch ein auf die Gerätespezifika des PDA angepasstes Interaktionsdesign. Hierbei wird eine zweihändige Bedienung mit unterschiedlichen Interaktionsformen eingesetzt, die bei geübten Nutzern keine visuelle Unterstützung am PDA-Bildschirm benötigt und somit blind eingesetzt werden kann. MOHOLT erhebt den Anspruch, sowohl intuitiv und leicht erlernbar zu sein, als auch eine gegenüber konventionellen PDA-Texteingabemethoden erhöhte Eingabeleistung zu ermöglichen. Eine erste Evaluation des Systems bestätigt diese Ansprüche weitgehend.

1 Einleitung

Die in den letzten Jahren zunehmend propagierte These des Zusammenwachsens verschiedener mobiler Kommunikations- und Informationssysteme (Magerkurth & Prante 2001) scheint sich mit jeder neuen Produktgeneration weiter zu bewahrheiten. Für Persönliche Digitale Assistenten (PDAs) ist das Bild des „verlängerten Arms“ für den Desktop-PC (Palm-OS Programmer's Companion 1999) längst überholt. Der bloße Informationsabruf ist einer umfangreichen Kommunikationsfunktionalität mit Multimedia-Inhalten und Telefonie gewichen. Ebenso erfüllt das Mobiltelefon mit Adressverzeichnissen oder Computerspielen immer mehr Funktionen eines PDA, wobei inzwischen in weiten Teilen Europas der *Short Message Service* (SMS) zum Austausch von Textnachrichten eine dominierende Stellung einnimmt.

Die Popularität der Texteingabe via Mobiltelefon hat eine große Forschungsaktivität hervorgerufen (z.B. Silfverberg et al. 2000; Bellman & MacKenzie 1998), da der Ziffernblock eines Telefons deutlich weniger Tasten bietet als das Alphabet Buchstaben enthält. Verschiedene Eingabesysteme wie *Multitap* oder *T9*® wurden entwickelt, um durch Mehrfachbelegung von Tasten das mobile Schreiben erst zu ermöglichen. Fortgeschrittene Systeme wie T9 arbeiten mit einem Abgleich von Worten in einem Wortverzeichnis (Silfverberg et al 2000). Geübte Benutzer erreichen inzwischen Geschwindigkeiten von 20 Worten pro Minute auf T9-Mobiltelefonen (James & Reichel, 2001).

Für PDAs, die im Gegensatz zu Mobiltelefonen mit berührungsempfindlichen, stiftgesteuerten Kleinstbildschirmen und einer deutlich geringeren Anzahl von Tasten ausgestattet sind, wurden zunächst Forschungsergebnisse von größeren stiftbasierten Systemen herangezogen, die entweder in der Tradition gestenbasierter Systeme oder virtueller Tastaturen (Masui 1998) stehen.

2 Texteingabe auf Persönlichen Digitalen Assistenten

Alle aktuellen PDA-Modelle, selbst scheckkartengroße Geräte wie der REX von Xircom, sind mit einer virtuellen Tastatur zur Texteingabe ausgestattet, zusätzlich bieten die größeren PDAs auch eine Art Handschrifterkennung.



Abbildung 1: Zeichenerkennung, Palm (links), virtuelle Tastatur, Palm (mitte), On-Screen-Handschrifterkennung, IPAQ (rechts)

Virtuelle Tastaturen (siehe Abbildung 1, mitte) bilden auf dem Bildschirm eine PC-Tastatur nach, wobei die Tasten direkt mit dem Stift gedrückt werden können. Die Funktionsweise virtueller Tastaturen ist damit unmittelbar ersichtlich und die Bedienung meist unproblematisch. Eine Wörterbuch-Unterstützung ähnlich den virtuellen Tastaturen auf größeren stiftbasierten Computern, z.B. POBox (Masui 1998), ist auch für PDAs wie den Compaq IPAQ verfügbar, allerdings nicht auf deutsch. Ein Dilemma ergibt sich einerseits aus der Forderung nach ergonomisch großen Tasten und andererseits einem möglichst geringen Platzbedarf auf dem Bildschirm. Da virtuelle Tastaturen auf PDAs nur mit einem statt zehn Fingern bedient werden können, lassen sich durch Aufgabe der üblichen QWERTZ(/Y)-Tastenanordnung z.T. beachtliche Geschwindigkeitssteigerungen erreichen (MacKenzie & Zhang 1999; siehe auch Perlin 1998).

Schrifterkennungssysteme für PDAs orientieren sich zumeist sehr eng an den *Unistrokes* (Goldberg & Richardson 1983) für stiftbasierte Rechner¹. Somit muss sich der Benutzer mit seiner Handschrift an die erkannten Gesten des Systems (siehe Abbildung 1, links) anpassen, was einen gewissen Lernaufwand, aber auch eine zuverlässigere Erkennung bedeutet. I.d.R. wird für die Zeicheneingabe ein Bereich unter oder auf dem Bildschirm (siehe Abbildung 1, rechts) reserviert, der im Gegensatz zu gestenbasierten PC-Systemen nicht verlassen werden darf.

3 Anforderungen an eine Hochleistungs-Texteingabe für PDAs

Die im vorigen Abschnitt vorgestellten Texteingaben weisen mehrere Unzulänglichkeiten auf, die einen Einsatz für bestimmte Benutzergruppen und Umgebungsbedingungen erschweren. Zunächst ist eine stiftbasierte Interaktion mit extrem kleinen Bildschirmen generell kein gutes Beispiel für eine gelungene Geräteadaptierung, da eine Fokussierung auf kleinste visuelle Merkmale hoher

¹ Im Dezember 2001 gewann Xerox, Entwickler von *Unistrokes*, einen Prozess gegen Palm Inc., deren Handschrifterkennung *Graffiti* aufgrund ihrer zu großen Ähnlichkeit die Patentrechte von Xerox verletzte.

Konzentration und motorischer Koordination bedarf. Gerade in „typischen“ PDA-Nutzungsbereichen wie Zugfahrten erschweren Umgebungsvibrationen die bildschirmorientierte Texteingabe. Folglich gilt als erste Anforderung an eine leistungsfähige Texteingabe:

⇒ *Der Anwendbarkeitsbereich soll möglichst groß sein.*

Analog zu einem großen Bereich an Umweltbedingungen, unter denen ein Einsatz möglich sein soll, gilt gleiches für den Menschen: Eine Texteingabe soll für möglichst viele Benutzer anwendbar sein. Dies beinhaltet vor allem die Frage nach der Erlernbarkeit. Während virtuelle Tastaturen hier praktisch keinen Lernaufwand benötigen, verlangen PDA-Zeichenerkennungssysteme wie Graffiti (Palm) mitunter erheblichen Übungsaufwand, da die erkannten Zeichen sich zwar meist an realen Schriftsätzen orientieren, jedoch selten mit ihnen identisch sind (siehe Abbildung 1, links). Als zweite Anforderung gilt daher:

⇒ *Eine Texteingabe soll leicht erlernbar sein.*

Um für mehr als nur einfachste Eingaben geeignet zu sein, muss eine mobile Texteingabe nach dem Erlernen eine ausreichende Eingabegeschwindigkeit ermöglichen. Es ist sicher unrealistisch, Eingaberaten wie bei PC-Tastaturen zu fordern, trotzdem bleibt Geschwindigkeit für den Benutzer ein wichtiges Kriterium; insbesondere gilt dies für Systeme, die beide Hände zur Bedienung beanspruchen und so keinen „Freihand-Bonus“ wie etwa beim Mobiltelefon bieten. Naturgemäß sind gestenbasierte Verfahren wie die Handschrifterkennung hier besonders problematisch, da das Anfertigen eines Zeichens prinzipiell langsamer ist als der Druck einer entsprechenden Taste auf einer (virtuellen) Tastatur. Es gilt drittens:

⇒ *Die Eingabegeschwindigkeit soll möglichst hoch sein.*

Eng verbunden mit der Forderung nach hoher Eingabegeschwindigkeit ist schließlich eine Robustheit der Eingabe. Eingabefehler können wiederum besonders bei einer Zeichenerkennung auftreten, da die Anforderungen an Gestalt und Geschwindigkeit der Zeichenanfertigung höher sind als das Treffen eines bestimmten Bildschirmbereiches mit einem Stift. Bei virtuellen Tastaturen treten Eingabefehler hingegen erst unter bestimmten Bedingungen auf, nämlich wenn entweder die Sicht auf den PDA-Bildschirm oder das Antippen selbst behindert werden. Für die Fehlerquote gilt somit nicht nur im Hinblick auf eine geringere Geschwindigkeit aufgrund von Selbstkorrekturen, sondern auch für die Akzeptanz bei den Benutzern:

⇒ *Die Fehlerquote soll möglichst gering sein.*

Eine Hochleistungs-Texteingabe muss allen vier Anforderungen (Anwendbarkeitsbereich, Erlernbarkeit, Geschwindigkeit, Fehlerzahl) genügen und damit Defizite etablierter Lösungen ausgleichen. Zusätzlich ist das Interaktionsdesign an die spezifischen Charakteristika des Persönlichen Digitalen Assistenten anzupassen.

4 Das MOHOLT-System

In den folgenden Abschnitten wird die „MOBILE HOCHLEISTUNGS-Texteingabe“ MOHOLT beschrieben, die die o.g. Anforderungen zu realisieren versucht. MOHOLT läuft auf PDAs mit dem Betriebssystem Palm-OS ab Version 3. Das Interaktionsdesign wurde angepasst an Geräte mit einer Taste an der Oberseite; dies beinhaltet die am weitesten verbreiteten Palm-Baureihen V und 500.

Im Gegensatz zu den auf PDAs verbreiteten Texteingabemethoden liegt MOHOLT eine Kombination aus tastenbasierter und displaygesteuerter Interaktion zugrunde. Der PDA wird zweihändig bedient, wobei mit der linken Hand die Tastenleiste und mit der rechten Hand vier virtuelle Tasten auf dem (berührungsempfindlichen) Bildschirm gesteuert werden. Die virtuellen Tasten lassen sich aufgrund ihrer Größe (je ca. $\frac{1}{4}$ des Bildschirms) blind betätigen, so dass eine bildschirmorientierte Interaktion i.S. einer Informationsaufnahme vom PDA-Bildschirm vermieden werden kann; damit wird der Anwendbarkeitsbereich gegenüber rein bildschirmorientierten Systemen vergrößert (s.o.). Zusätzlich kann mit der oberen Handfläche der rechten Hand eine Taste an der Geräteoberseite bedient werden (siehe Abbildung 2, rechts).

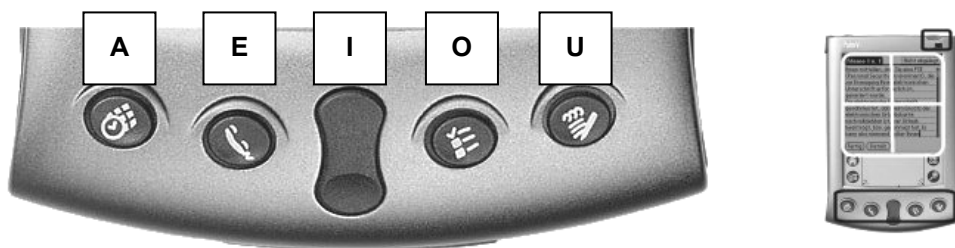


Abbildung 2: Zuordnung der Vokale zu den Tasten (links), Tastenverteilung (rechts)

Die verschiedenen Tasten werden kombiniert zur Zeichengenerierung verwendet, wobei aufgrund der geforderten hohen Eingabegeschwindigkeit keine Interaktionssequenzen nötig sind, d. h. auf eine Mehrfachbetätigung von Tasten zur Erstellung eines einzelnen Zeichens kann verzichtet werden. Bei der Kombination verschiedener Tasten wird auf ein Ankerkonzept zurückgegriffen, das im nächsten Abschnitt erläutert wird.

4.1 Ankerkonzept der Zeichengenerierung

Um der Anforderung einer schnellen Erlernbarkeit gerecht zu werden, wurde bei der Tastenbelegung ein Ankerkonzept entworfen, das beim Benutzer bereits vorhandenes Wissen verwendet und als Ankerpunkte bei der Erlernung der vollständigen Funktionalität des Systems einsetzt. Vorausgesetzt wird die Folge der Vokale „A-E-I-O-U“, die als überlernt gelten kann. Diese Vokalfolge wird von links nach rechts den (Hardware-)Tasten des PDAs zugeordnet, so dass ein Druck auf die Taste ganz links ein „A“ generiert und analog die Taste ganz rechts ein „U“ (siehe Abbildung 2). Die Konsonanten werden nun entsprechend ihrer Position im Alphabet den Vokalen zugeordnet. Hierbei gilt die Regel, dass alle Konsonanten, die einem Vokal nachfolgen, diesem zugehörig sind (siehe Tabelle 1). Dies ermöglicht auch dann eine intuitive Lokalisierung von Konsonanten, wenn das Alphabet beim Benutzer nur sequentiell abrufbar ist. Selbst wenn der Benutzer also nicht genau weiss, wann im Alphabet ein bestimmter Konsonant auftaucht, kann er diesen anhand des Ankervokals ohne größeren Aufwand zumindest ungefähr bestimmen. Ein „F“ ließe sich somit leicht der Taste „E“ zuordnen, ebenso wie das „Y“ aufgrund des Wissens um seine ungefähre Position dem „U“ zugeordnet würde.²

² Rückmeldungen von Benutzern beim Erlernen von MOHOLT heben hier besonders die häufige Richtigkeit spontaner, nicht überlegter Zuordnungsversuche hervor.

4.2 Interaktionsformen

Wie oben beschrieben, werden Vokale durch einfaches Betätigen der entsprechenden Tasten erstellt. Zur Generierung von Konsonanten wird zusätzlich zu der Taste des (Anker)-vokals auf dem Bildschirm eine virtuelle Taste oder die Hardware-Taste an der Gehäuseoberseite gedrückt. Dies entspricht dem Vorgehen mit Umschalt-, Steuerung- oder Alt-Tasten an PC-Tastaturen. Tabelle 1 zeigt die Tastenzuordnung.

Taste (unten)	Vokal	Virtuelle Tasten				Taste (oben)
	A	B	C	D		
	E	F	G	H		
	I	J	K	L	M	N
	O	P	Q	R	S	T
	U	V	W	X	Y	Z

Tabelle 1: Aufteilung der Vokale und zugehöriger Konsonanten

Für die Betätigung der Vokal-Tasten, der virtuellen Tasten und der Taste an der Gehäuseoberseite werden drei verschiedene Interaktionsformen eingesetzt, die in Abbildung 3 erläutert werden. Die Einführung der verschiedenen Interaktionsformen dient durch das Prinzip des *Chunking* wiederum der Erlernbarkeit, da für keine einzelne Interaktionsform mehr als vier Elemente gelernt werden müssen.

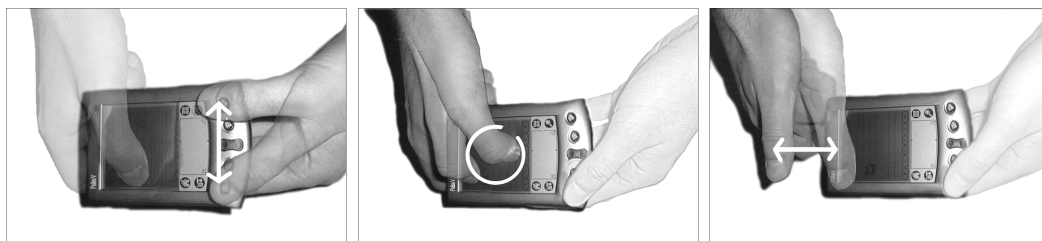


Abbildung 3: Drei Interaktionsformen: Tastendruck mit Daumen, linkshändig (links), Bildschirmberührung mit Daumen, rechtshändig (mitte), Tastendruck mit Handfläche, rechtshändig (rechts)

Von diesem Prinzip abgedeckt wird auch die Generierung vergleichsweise seltener Zeichen wie Zahlen, Satz-, Sonder- oder „Korrekturzeichen“. Diese werden durch Kombination mehrerer Vokal-Tasten erstellt, wobei es hier keine Lernerleichterung durch ein Ankerkonzept gibt. Darüber hinaus verlangen diese selteneren Zeichen ein Umgreifen mit der rechten Hand, um simultane Tastenbetätigungen an der unteren Tastenleiste vornehmen zu können. Wie aus der Praxis von Benutzerstudien ersichtlich (Dunlop & Crossan 2000; James & Reischel 2001) lässt sich bei vielen Anwendungsbereichen auf diese selteneren Zeichen auch gänzlich verzichten, so dass eine Optimierung der Erstellung von Buchstaben auf Kosten seltener Sonderzeichen sinnvoll erscheint.

4.3 Rückmeldung über generierte Zeichen

Da MOHOLT ohne eine bildschirmbasierte Informationsaufnahme eingesetzt werden kann, steht eine akustische Rückmeldung über generierte Zeichen zur Verfügung, die für jedes Zeichen die Tonhöhe sowie für jede Gruppe aus Vokal und zugeordneten Konsonanten die Tonlänge variiert. Zusätzlich können auch visuelle Rückmeldungen erfolgen, die besonders beim Erlernen des Systems sinnvoll sind. Diese werden im folgenden Abschnitt erläutert.

4.4 Software-Realisierung

Für MOHOLT wurden mehrere Software-Komponenten entwickelt. Zunächst steht mit MOHOLT-Tutor (siehe Abbildung 4, links) eine Applikation zur Verfügung, die zum Erlernen der Texteingabe dient. Hierbei werden auf dem Bildschirm Handlungsanweisungen an den Benutzer ausgegeben, durch deren Befolgung das System verstanden werden soll. Auf dem Bildschirm werden sowohl die virtuellen Tasten als auch die *Hard-Keys* visualisiert, wobei sich die Darstellung in Abhängigkeit der gedrückten Anker-Tasten anpasst. Nach der Erstellung eines Zeichens wird dieses zusammen mit einem Signalton am Bildschirm ausgegeben.

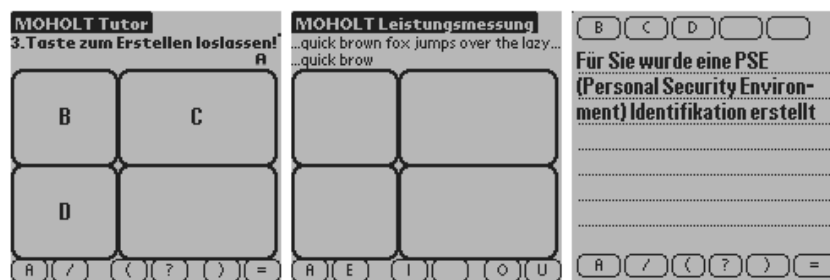


Abbildung 4: MOHOLT-Komponenten: Tutor (links), Leistungsmessung (mitte), Texteditor (rechts)

Die Komponente zur Leistungsmessung (siehe Abbildung 4, mitte) gibt dem Benutzer Sätze zur Eingabe vor und misst dabei Zeit- und Fehlerzahl. Auch hierbei wird eine visuelle und optional eine akustische Unterstützung des Benutzers realisiert. Am Ende einer Leistungsmessung wird neben den erfassten Daten eine deskriptive Statistik über Mittelwerte und Varianz auf Satzebene ausgegeben.

Zur Verwendung in eigenen Applikationen existiert schließlich eine C++-Klasse, die die MOHOLT-Funktionalität kapselt und neben der akustischen Rückmeldung bei Bedarf eine kleine *On-Screen*-Hilfe am Bildrand einblenden kann. Durch die verkleinerte Darstellung wird dabei ein übermäßiger Platzbedarf, wie er bei virtuellen Tastaturen entsteht, vermieden. Abbildung 4, rechts, zeigt einen einfachen Texteditor, bei dem die Texteingabe mittels MOHOLT geschieht.

5 Benutzerstudie

Obwohl mehrere, teils sehr komplexe mathematische Modelle zur Vorhersage von Texteingabeschwindigkeiten existieren, ist „neither model particularly accurate“ (James & Reischel 2001).

Darüber hinaus ist die Annahme wenig sinnvoll, dass es sich bei den Benutzern um Experten handelt, die keine Fehler bei der Eingabe machen. Folglich ist zur Abschätzung der Anwendbarkeit einer Texteingabe eine Benutzerstudie unerlässlich.

Eine erste Evaluation von MOHOLT hat bereits stattgefunden, das experimentelle Design wurde hierbei weitgehend der Untersuchung von James und Reischel (2001) entnommen, Details können dort nachgelesen werden. Im folgenden wird eine kurze Darstellung der wesentlichen Ergebnisse gegeben.

5.1 Methode

5.1.1 Teilnehmer

30 Studenten einer Hochschule ($m = 24$, $w = 6$, $\bar{X} 28,4$ Jahre, keine PDA-Benutzer) wurden in drei Gruppen unterteilt. Jede Gruppe wurde einer PDA-Texteingabe zugeordnet (MOHOLT-Leistungsmessung, Virtuelle Tastatur, Graffiti-Schrifterkennung). Jeweils die Hälfte der Teilnehmer konnte die zugeordnete Texteingabe intensiv trainieren („Experten“), während die andere Hälfte nur eine Demonstration und eine fünf-minütige Aufwärmphase erhielt („Anfänger“).

5.1.2 Aufbau und Durchführung

Jedem Teilnehmer wurden hintereinander zehn Sätze vorgegeben, die möglichst schnell und ohne Selbstkorrektur von Fehlern mit einem Palm-PDA eingegeben werden mussten. Die Sätze wurden der Untersuchung von James und Reischel ohne Übersetzung entnommen, um so einen Vergleich zu den Eingabemethoden der dort verwendeten Mobiltelefone herstellen zu können. Die eine Hälfte der Sätze bestand aus typischen informellen Kurznachrichten (z.B. „*let me know if we should wait*“), die andere aus längeren, „seriösen“ Sätzen (z.B. „*the slinky and the jump rope are among five toys that will be designated wednesday*“). Satz- und Sonderzeichen wurden nicht verwendet. Zeitdauer und Fehler wurden für jeden Satz direkt auf dem Untersuchungs-PDA erhoben.

5.2 Ergebnisse

Für die Auswertung wurde ein 3 (Texteingabe) $\times 2$ (Erfahrung) $\times 2$ (Textart) -Design gewählt, wobei die gemittelten Eingabegeschwindigkeiten in Tabelle 2 und die Gesamtzahl der Fehler pro Gruppe in Tabelle 3 aufgeführt sind. Ergebnisse für Mobiltelefone aus James & Reischel (2001) sind jeweils zum Vergleich mit angegeben.

Signifikante Haupteffekte der Varianzanalysen wurden für die Eingabegeschwindigkeit als abhängige Variable bei der Erfahrung ($F(2, 151) = 11.7$, $p < 0.5$) sowie für die Gesamtfehlerzahl bei der PDA-Texteingabe ($F(2, 151) = 10.8$, $p < 0.5$) gefunden. Ein Interaktionseffekt wurde für die Eingabegeschwindigkeit als abhängige Variable bei Erfahrung $\times$ PDA-Texteingabe ($F(1, 151) = 12.3$, $p < 0.5$) gefunden.

Die Aussagekraft der deskriptiven Vergleiche zwischen Texteingaben für PDAs und Mobiltelefonen wird beeinträchtigt durch den Umstand, dass bei James & Reischel (2001) auch SMS-„Anfänger“ Geräteerfahrung mitbrachten und lediglich durch eine geringere Anzahl verschickter SMS-Nachrichten pro Woche identifiziert wurden.

Textart	Erfahrung	MOHOLT	Virtuelle Tastatur	Zeichen-erkennung		T9*	Multitap*
Kurz	Anfänger	14,62	19,93	4,6		10,98	10,37
	Experte	24,5	31,43	15,42		25,68	10,53
Länger	Anfänger	12,58	14,15	4,33		7,21	5,59
	Experte	21,27	26,61	12,64		15,05	5,33

* aus James & Reischel 2001

Tabelle 2: Gemittelte Eingabegeschwindigkeit (Worte pro Minute³)

Textart	Erfahrung	MOHOLT	Virtuelle Tastatur	Zeichen-erkennung		T9*	Multitap*
Kurz	Anfänger	24	25	99		24	31
	Experte	21	24	53		17	17
Länger	Anfänger	26	35	147		20	85
	Experte	20	29	88		17	48

* aus James & Reischel 2001

Tabelle 3: Gesamtfehlerzahl pro Gruppe

5.3 Diskussion

Während die Geschwindigkeit von MOHOLT deskriptiv nicht ganz an die einer virtuellen Tastatur heranreicht, liegt sie sowohl deutlich über der einer Zeichenerkennung als auch über denen der beiden Texteingaben für Mobiltelefone. Dies spricht für eine Erfüllung der Forderung nach hoher Eingabegeschwindigkeit. Die geringen Unterschiede in der Fehlerzahl zwischen wenig geübten Benutzern und „Experten“ deuten auf eine besonders leichte Erlernbarkeit von MOHOLT hin. Die absolute Fehlerzahl sowohl für „Experten“ als auch für „Anfänger“ spricht für die geforderte Robustheit von MOHOLT.

Die Texteingaben für Mobiltelefone weisen im Vergleich zu den PDA-Systemen zwar relativ geringe Fehlerraten auf; hier muss allerdings darauf hingewiesen werden, dass das *Response-Bias*, also die individuelle Neigung der Probanden, eher eine niedrige Geschwindigkeit oder eine große Fehlerzahl zu riskieren, nicht erfasst wurde. In der Untersuchung von James & Reischel könnte daher der Geschwindigkeitsaspekt eine geringere Rolle gespielt haben, zumal auf PDAs z.T. deutlich höhere Geschwindigkeiten erreicht wurden.

³ In Anlehnung an James & Reischel (2001) wird eine durchschnittliche Anzahl von 5.98 Zeichen pro Wort zugrundegelegt. Deutsche Worte sind vermutlich länger.

6 Anwendungsbeispiel

MOHOLT wird bereits in ersten Prototypen sogenannter *ServiceClients* eingesetzt. Diese werden als PDA-Systeme implementiert, deren Softwareausstattung eine Adaptierung von Benutzungsoberfläche und Funktionalität in Abhängigkeit von Artefakten der Umgebung und Charakteristika des Benutzers realisiert. Somit können mobile Kleingeräte wie PDAs oder Mobiltelefone in einem *Smart Environment* Dienste für andere Artefakte anbieten (vgl. Beigl 1999).

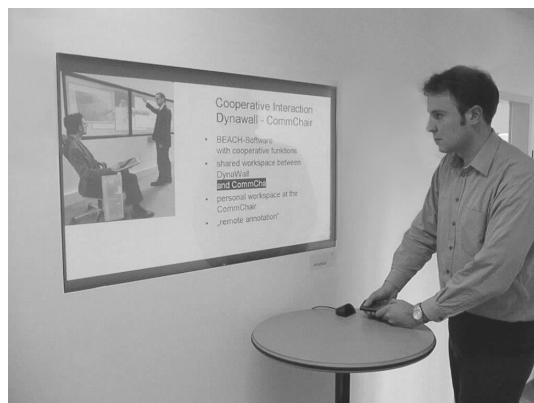


Abbildung 5: MOHOLT auf einem ServiceClient

Abbildung 5 zeigt einen PDA-ServiceClient, der zur Edierung eines elektronischen Posters auf dem Gang einer Smart-Office-Umgebung eingesetzt wird. Der PDA stellt eine Infrarotverbindung zu dem IR-Sensor der Wandtafel her und erhält als persönliches Gerät eines bestimmten Benutzers („Service-Techniker“) die Berechtigung, mittels MOHOLT den Text des Posters auf der (passiven) Wandtafel zu verändern. Die visuelle Rückmeldung wird hierbei direkt auf der großen Wandtafel anstatt auf dem kleinen PDA-Bildschirm gegeben. Dies ermöglicht eine natürlichere und ergonomischere Interaktion als mit traditionellen PDA-Texteingabesystemen. Der PDA wird zur mobilen Tastatur, die ohne eigenes Display auskommt.

7 Ausblick

In diesem Beitrag wurde eine neuartige Texteingabe für PDAs, MOHOLT, vorgestellt, der eine Kombination aus tastenbasierter und displaygesteuerter Interaktion zugrunde liegt. Nachdem sich erste Software-Realisierungen in einer Benutzerstudie als viel versprechend erwiesen haben, werden sich zukünftige Entwicklungen aufgrund der Forderungen nach Robustheit und großem Anwendbarkeitsbereich weiter an der Unterstützung der durch MOHOLT ermöglichten blinden Bedienbarkeit orientieren. Dazu gehört v.a. eine Erweiterung der akustischen Rückmeldungen, die etwa auf Wortebene bekannte oder unbekannte Worte unterschiedlich kennzeichnen können.

Überdies werden im Rahmen von Smart Environments weitere Szenarien zu entwickeln sein, in denen in Interaktion mit umgebenden Artefakten textbezogene Dienste über MOHOLT angeboten werden können.

8 Danksagung

Thorsten Prante und Norbert Streitz danke ich sehr für ihre wertvollen Hinweise sowie Sascha Nau für die Illustrationen und Torsten Holmer für die Fotografien. Diese Arbeit wurde durch das Ladenburger Kolleg „Living in a smart environment“ der Daimler-Benz-Stiftung finanziert.

Literatur

- Beigl, M. (1999): Point & Click - Interaction in Smart Environments. In: Gellersen, H.-W. (Hrsg.): Lecture notes in computer science; Vol 1707, Heidelberg: Springer, 311-314.
- Bellman, T.; MacKenzie, I. S. (1998): A probabilistic character layout strategy for mobile text entry. In: *Proceedings of Graphics Interface '98*. 168-176.
- Dunlop, M. D.; Crossan, A. (2000): Dictionary based text entry method for mobile phones. In: *Second Workshop on Human-Computer Interaction with mobile devices*, 5-7.
- Goldberg, D.; Richardson, C. (1993): Touch-Typing with a stylus. In: *Proceedings of the conference on human factors in computing systems 1993*, 80-87.
- James, C. L. ; Reischel, K. M. (2001): Text input for mobile devices: Comparing model prediction to actual performance. In: *Proceedings of CHI'2001*, New York: ACM, 365-371.
- Magerkurth, C; Prante, T (2001): Towards a unifying approach to mobile computing. *SIGGROUP Bulletin*, April 2001, 22, 16-18.
- MacKenzie, I. S.; Zhang, S. X. (1999): The Design and Evaluation of a High-Performance Soft Keyboard: *Proceedings of CHI'99*, New York: ACM, 15 – 20.
- Masui, T. (1998): An Efficient Text Input Method for Pen-based Computers. In: *Proceedings of CHI'98*, New York: ACM, 328 – 335.
- Palm-OS Programmer's Companion (1999): <http://www.palmos.com>
- Perlin, K. (1998): Quikwriting: Continuous Stylus-based Text Entry. In: *Proceedings of UIST'98*, New York: ACM, 215-216.
- Silfverberg, M.; MacKenzie, I.S.; Korhonen, P. (2000): Predicting text entry speed on mobile phones. In: *Proceedings of CHI'2000*, New York: ACM, 9-16.

Kontaktinformationen

Carsten Magerkurth
Fraunhofer IPSI
Dolivostraße 15
64293 Darmstadt
magerkurth@ipsi.fhg.de