

# Interaktionen blinder Nutzer bei der Bedienung linearisierter Oberflächen

Henrik Eichstädt

Technische Universität Ilmenau

## Zusammenfassung

Der vorliegende Beitrag entwickelt, ausgehend von theoretischen Betrachtungen, Strategien blinder Nutzer bei der Bedienung HTML-basierter Anwendungsoberflächen. Die Strategien werden in einer an das GOMS-Modell angelehnten Form beschrieben. Es wird gezeigt, dass ein GOMS-Modell für die Beschreibung der Strategien geeignet ist. Ziel ist es, durch die Anwendung des GOMS-Modells die Berechnung der Ausführungsdauer von Nutzerinteraktionen als Effizienzmaß zu nutzen. Dieses Usability-Kriterium kann schon während der Designphase für die Bewertung der Benutzbarkeit von HTML-basierten Applikationsoberflächen durch blinde Nutzer verwendet werden. In Nutzertests, die dieser theoretischen Arbeit folgen, sollen die vorhergesagten Strategien verifiziert werden.

## 1 Einleitung

Eine Schwierigkeit bei der Entwicklung von Nutzeroberflächen ist die Vorhersage, ob die angestrebte Oberfläche Usability-Kriterien gerecht wird. Es gibt kaum harte Fakten, an denen eine Oberfläche ausgemessen werden kann. Durch quantitative Usability-Tests können Indikatoren berechnet werden, die einen Rückschluss auf die Usability einer Oberfläche zulassen. Qualitative Usability-Tests zeigen Probleme der Nutzer bei der Interaktion mit der Oberfläche auf. Beide Varianten sind aber in der Regel erst zuverlässig anwendbar, wenn schon eine brauchbare Oberfläche in Form eines mehr oder weniger interaktiven Prototyps vorliegt.

Dem gleichen Problem begegnen zunehmend Entwickler, die barrierefreie Oberflächen erzeugen möchten. Bestehende Richtlinien wie die Web Content Accessibility Guidelines (‚WCAG‘, Chisholm et al. 1999), die Authoring Tool Accessibility Guidelines (‚ATAG‘, Treviranus et al. 2000) oder die deutsche „Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz“ (‚BITV‘, BMGS 2002) erläutern die gegebenen Prüfpunkte. Konkrete Handlungsanweisungen werden den Entwicklern von

Anwendungsoberflächen allerdings vorenthalten. In der Praxis werden die Richtlinien teilweise sehr individuell interpretiert. Eine ‚richtige‘ Lösung scheint es nur selten zu geben<sup>1</sup>.

Auch bezüglich Barrierefreiheit bleiben letztlich ‚nur‘ Usability-Tests unter Berücksichtigung der entsprechenden Zielgruppen. Diese Tests sind aber - aufgrund der Vielzahl zu betrachtender Einschränkungen bei den Nutzern - sehr aufwändig. Robertson et al. (2003) beschrieben die Notwendigkeit eines mobilen Testlabors, um die Testpersonen an Ihren gewohnten, an die jeweiligen Einschränkungen angepassten Arbeitsplätzen beobachten zu können. Die betrachteten Testpersonen stellten dennoch nur eine sehr kleine Stichprobe der zu berücksichtigenden Nutzer dar. In der Praxis werden Tests - nicht zuletzt wegen des wirtschaftlichen und zeitlichen Drucks - bei einigen Projekten remote in der Form von Kurztests mit über Mailinglisten rekrutierten, freiwilligen Experten durchgeführt<sup>2</sup>.

Automatisierte Überprüfungen der Barrierefreiheit von Oberflächen sind nur in Teilen möglich. Verschiedene Dienste sind verfügbar, die anhand der ‚harten‘ Prüfpunkte evaluieren, ob HTML-Code den Kriterien entspricht. Es handelt sich aber nur um eine Überprüfung auf rein technischer Ebene. Viele wichtige Barrierefrei-Kriterien erfordern allerdings eine inhaltliche Überprüfung, die bisher nicht automatisiert vorgenommen werden kann.

Anwendungen wie Web Content Management Systeme oder Webmailer stellen oft eine HTML-basierte grafische Nutzeroberfläche zur Verfügung. Für die Interaktion blinder Nutzer mit solchen, durch einen Webbrowser präsentierten Oberflächen werden in diesem Beitrag Nutzungsstrategien aus Expertenwissen, Erfahrungsberichten von Nutzern sowie durch in den Richtlinien aufgeführten Empfehlungen entwickelt und als GOMS-Modell beschrieben.

Das Modell soll als Grundlage für die Berechnung der Ausführungsdauer zur Lösung bestimmter Nutzeraufgaben dienen. Im Rahmen der Usability-Bewertung einer Oberfläche kann die Ausführungsdauer bereits begleitend zum Oberflächen-Design als Maß für die Benutzbarkeit / Usability verwendet werden. Designalternativen lassen sich so hinsichtlich der Effizienz der Oberflächennutzung vergleichen.

Das theoretisch entwickelte Modell wird zu einem späteren Zeitpunkt durch Nutzertests verifiziert werden. Diese Tests finden erst nach der Fertigstellung des vorliegenden Beitrags statt. Die Verifikation des Modells durch Nutzerbeobachtungen konnte in diesem Beitrag demnach noch nicht behandelt werden. Im Rahmen der Nutzertests werden durch blinde und sehgeschwache Nutzer per Tastatur bediente und über das Gehör bzw. den Tastsinn wahrgenommene Oberflächen betrachtet.

---

<sup>1</sup> Entsprechende Diskussionen sind regelmäßig auf verschiedenen Mailinglisten und in Newsgroups zu beobachten. Z.B. die WAI-de Liste des Fraunhofer Institut für Angewandte Informationstechnik (FIT, <http://access.fit.fraunhofer.de/waideinfo?lang=de>) oder die Yahoo-Group ‚uvip - Usability for visually impaired people‘ (<http://groups.yahoo.com/group/uvip/>).

<sup>2</sup> Neben der WAI-de Liste z.B. auch in der Yahoo-Group ‚uvip-web-test · Web page testing by visually impaired people‘ (<http://groups.yahoo.com/group/uvip-web-test/>)

## 2 Barrierefreiheit

Die Anforderungen, die aus Spezifikationen zur Barrierefreiheit (BITV, WCAG, ATAG) resultieren, haben einen starken Bezug auf Behinderungen. Oft wird aus den Augen verloren, dass die Berücksichtigung von Barrierefreiheit auch allen anderen Nutzer – solchen mit veralteter Hardware, im Alter auftretenden Gebrechen oder auch Neulingen in einem Anwendungsgebiet – zugute kommt. Prinzipiell bedeutet Barrierefreiheit, dass kein Nutzer von einem ‚Angebot‘ ausgeschlossen wird. Ein Angebot kann der Zutritt zu einer Gaststätte, die Benutzung eines Geldautomaten oder der Besuch einer Website sein. Bezogen auf Informationstechnologie wirken die zur Verfügung stehende Hardware, das Betriebssystem, ein Programm / eine Website (bzw. deren Oberfläche) auf die Barrierefreiheit eines Angebots ein und können - wie Behinderungen - eine Einschränkung bedeuten.

### 2.1 Eingrenzung der Zielgruppe

Es ist unmöglich, alle vorstellbaren Einschränkungen im Rahmen einer zeitlich begrenzten Arbeit umfassend zu betrachten. Zumal die denkbaren (und auftretenden) Kombinationen von Einschränkungen die Vielfalt der potentiellen Zielgruppen erhöhen. Die hinsichtlich Barrierefreiheit zu betrachtende Zielgruppe wird daher für die nachfolgende Betrachtung und die Fortführung des Forschungsvorhabens auf ein sinnvolles Maß reduziert.

Aus zwei Gründen wird der Fokus dieser Arbeit auf blinde Nutzer gesetzt. Zum einen setzen sich die verfügbaren offiziellen Richtlinien in großem Umfang mit den Anforderungen blinder Nutzer auseinander<sup>3</sup> und weisen somit Mängel bei Anforderungen anderer Einschränkungen auf. Diese Richtlinien bilden aber einen Teil des theoretischen Grundstocks dieser Arbeit. Somit ist es vernünftig, einen Fokus zunächst auf die dort am meisten berücksichtigte Nutzergruppe zu legen.

Zum anderen ist es gerade bei blinden Nutzern notwendig, eine andere Art der Oberfläche zu berücksichtigen. Während viele Behinderungen weiterhin die Arbeit mit einer visuellen Oberfläche erlauben, müssen sich blinde Nutzer auf eine durch das Gehör oder den Tastsinn zu erfassende Oberfläche stützen. Diesen Oberflächen liegen andere Gestaltungskonzepte zugrunde, die andere (Inter)aktionsstrategien auf der Nutzerseite erfordern.

### 2.2 Barrierefreiheit und blinde Nutzer

Um als Nutzer auf nichtvisuelle Oberflächen zugreifen zu können werden Hilfstechnologien (Screenreader) und spezielle Ausgabegeräte (Audioausgabe, Braillezeile) benötigt. Diese Technologien benötigen geeignet aufbereiteten Content, um eine sinnvolle Ausgabe erzeu-

---

<sup>3</sup> Ein großer Kritikpunkt an der BITV ist genau der Umstand, dass sich die Bedürfnisse blinder Nutzer wieder finden, andere Behindertengruppen aber nicht in adäquatem Umfang berücksichtigt werden. Es wird erwartet, dass die in Aussicht stehende BITV 2 in dieser Hinsicht Verbesserungen aufweist.

gen zu können. Visuelle Strukturierungen und grafische Elemente werden durch diese Ausgabeformen nicht oder nur rudimentär berücksichtigt: „Alternative Ausgabegeräte [...] erfordern die Beachtung einer textorientierten und linearisierbaren Informationsgestaltung“ (Hellbusch 2004, S. 7).

Insbesondere für Oberflächen von Serverapplikationen bedeutet dies, dass die Elemente der einzelnen, linearisiert ausgegebenen Seiten eine sinnvolle und auswertbare Struktur aufweisen müssen. Anhand der logischen Struktur kann die unterstützende Hilfstechnologie Filter anbieten, um den Nutzer beim Auffinden relevanter Inhalte zu unterstützen. Die Benutzung linearisierbarer Seiten ist eine ideale Voraussetzung für die erfolgreiche Anwendung des GOMS-Modells.

### 2.3 Funktionen von Screenreadern

Um die Aufgaben adäquat beschreiben zu können, müssen zunächst die für die Zielstellung relevanten, von Screenreadern angebotenen Funktionalitäten betrachtet werden. Es ist zu beachten, dass abhängig vom genutzten Screenreader die Funktionalitäten nicht oder nur eingeschränkt zur Verfügung stehen können. Die Beschreibung der Funktionalitäten orientiert sich an einem der meistgenutzten Programme – Jaws von Freedom Scientific<sup>4</sup>.

- *Orientierungsphase.* Wird eine neue HTML-Seite geladen, beginnt das Ausgeben der Seite mit der Nennung des Seitentitels sowie der Angabe über das Vorkommen von Links, Überschriften und weiteren besonderen Elementen („Seite hat 8 Überschriften und 42 Links“).
- *Tastaturbedienung.* Da blinde Nutzer nicht mit Zeigegeräten auf Elemente der Oberfläche zeigen können, erfolgt die Bedienung ausschließlich mit Hilfe von Tastaturkommandos, die erinnert werden müssen. Bei einer reinen Sprachausgabe findet daher kein Wechsel auf ein Zeigegerät (z.B. eine Maus) statt. Werden die Inhalte auf einer Braillezeile für die haptische Wahrnehmung ausgegeben, wechselt die Handposition zwischen Braillezeile und Tastatur.
- *Inhaltlicher Fokus = (Inter)Aktionsfokus.* Während sehende Nutzer z.B. Inhalte eingeben, parallel aber einen ganz anderen Teil der Oberfläche wahrnehmen können, fallen diese beiden Fokusse bei blinden Nutzern zusammen. Der Ort, an dem agiert wird ist gezwungenermaßen auch der Ort, der ausgegeben und wahrgenommen wird. Dieses Funktion stützt die These, dass GOMS für die Analyse geeignet ist.
- *Ansage der Elementart.* Gelangt der Screenreader während des Vorlesens an ein besonderes Element, so wird zusätzlich zum Inhalt des Elements auch die Art des Elements ausgegeben („Link:“, „Grafik:“). Während die Dauer für die Ansage der Elementart konstant ist, variiert die Dauer für die Ansage des Inhalts mit dem Umfang der Textinformation.

---

<sup>4</sup> <http://www.freedomsci.de/> bzw. <http://www.freedomsci.com/>

- *Vorlesemodus wechseln.* Abhängig von den in der Seite vorkommenden Inhalten kann der Ausgabemodus gewechselt werden (z.B. Formular-, Tabellen oder normaler Modus). So werden die entsprechenden Strukturen bei der Ausgabe optimal berücksichtigt
- *Zwischen HTML-Elementen springen.* Während Inhalte ausgegeben werden, kann der Nutzer die Ausgabe abbrechen und den Fokus auf andere Elemente der Seite setzen, um Textteile zu überspringen. So ist es auch möglich, nur Aufzählungen, Textabsätze oder Überschriften zu berücksichtigen. Für das Springen zu Überschriften oder Links gibt es spezielle Tastenkommandos.

## 3 GOMS

Das GOMS-Modell wurde Anfang der 80er Jahre von Card, Moran und Newell eingeführt und als ingenieurtechnisches Modell zur Bewertung von Oberflächen beschrieben. Card et al. betrachteten fast ausschließlich verschiedenste Texteditoren (Card et al. 1983). Andere Forschungsarbeiten wendeten das Modell später auf Tabellenkalkulationen, Zeichenprogrammen und andere Arbeitsbereiche an. Alle Betrachtungen zeigten, dass GOMS für die Vorhersage von Zeiten, die zur Ausführung einer bestimmten Aufgabe benötigt wurden, geeignet ist und als Werkzeug bei der Entwicklung von Anwendungen dienen kann. (z.B. John, Kieras 1996)

### 3.1 Voraussetzungen für GOMS

Für die Anwendung von GOMS sind vier Voraussetzungen zu erfüllen:

- *Erfahrende Nutzer.* GOMS-Analysen erfassen nur diejenigen Nutzer hinreichend gut, die mit der betrachteten Anwendungsoberfläche vertraut sind. Novizen oder Gelegenheitsnutzer kann das Modell nur mit Einschränkungen erfassen.
- *Linearisierte Abläufe.* Eine weitere Voraussetzung ist, dass die betrachteten Abläufe weitestgehend durch sequentielle Abfolgen abgebildet werden können<sup>5</sup>.
- *Fehlerfreie Ausführung.* GOMS-Analysen gehen davon aus, dass der erfahrene Nutzer die Aufgaben weitgehend fehlerfrei durchführt.
- *Geschlossen Aufgabe.* Bei den betrachteten Aufgaben muss ein konkretes Ziel vorgegeben sein, das über einen spezifizierten Lösungsweg erreicht werden soll.

---

<sup>5</sup> Die parallele Ausführung wird in einer eigenen GOMS-Spezifikation, dem CPM-GOMS berücksichtigt. ‚CPM‘ steht dabei sowohl für ‚Cognitive-Perceptual-Motor‘ als auch ‚Critical-Path Method‘ (John, Gray 1995)

## 3.2 Anwendung von GOMS

„GOMS“ steht für *Goals, Operators, Methods* und *Selection rules*. Im Fokus eines GOMS-Modells steht die Zielstellung (Goal), die durch das Ausführen bestimmter Methoden (Methods) erreicht werden kann. Stehen zur Erreichung eines Ziels mehrere Methoden zur Verfügung, ist es notwendig Regeln zu definieren, die eine Entscheidung über die Anwendung der jeweils am Besten geeigneten Methode ermöglichen (Selection rules). Die „atomaren Teilchen“ in diesem Konstrukt sind Operatoren (Operators), die keine weitere Verzweigung mehr aufweisen. Jeder Operator und jede Methode stellt ein eigenes Teilziel dar. Den Methoden sind Operatoren und weitere Methoden untergeordnet.

Aufgrund der aufgestellten (Unter)Methoden, Operatoren und Auswahlregeln ist es möglich, die zum Erreichen des Ziels notwendigen Abläufe zu beschreiben. Dies geschieht in einer Programmiersprachen ähnlichen Form. Den Operatoren, der kleinsten möglichen Einheit in diesem „Skript“ sind Zeiten (empirisch ermittelt oder durch Gesetzmäßigkeiten beschrieben) zugeordnet, die die Ausführungsdauer jedes Operators kennzeichnen. Für das Erreichen eines Ziels werden die einzelnen Zeiten für die Abarbeitung aller notwendigen Operatoren addiert. So ergibt sich die Ausführungsdauer („execution time“), die zum Erreichen des Ziels benötigt wird. Die Ausführungsdauer stellt somit ein Effizienzmaß dar und kann zur Bewertung verschieden ausgeführter Oberflächen verwendet werden um diese zu vergleichen.

Card et al. führen sechs Basis-Operatoren ein, um Nutzeraktionen hinreichend zu beschreiben (Card et al. 1983). Kieras erweitert die im ursprünglichen Modell definierten Operatoren und passt diese an (Kieras 2001). Jeder Operator repräsentiert z.B. einen Tastendruck, eine Mausebewegung oder das Platzieren der Hand auf einem Eingabegerät.

Für die vorliegende Arbeit werden die Operatoren K (Keypress, eine Taste drücken), W(t) (Wait, die Zeit, die der Nutzer auf eine Reaktion des Systems warten muss), T(n) (Type a sequence, n Tastaturanschläge für einen chunk) und M (Mental, für mentale Abläufe) benötigt. Die Betrachtung des Operators H (Homing, das Platzieren der Hand auf einem Gerät) wird zunächst zurückgestellt. Die Operatoren P (Point), B (Mousebutton) und D (Draw) sind für die in diesem Beitrag vorgenommenen, auf nicht visuell wahrgenommene Oberflächen bezogene Betrachtungen nicht relevant.

## 4 Aufgabenmodelle

Nachfolgend werden die drei grundlegenden Aufgaben, die bei der Bedienung von HTML-basierten Oberflächen benötigt werden, entwickelt und als GOMS-Modell beschrieben. Für die Betrachtung des Nutzerverhaltens blinder Nutzer wird eine gemäß BITV und Best Practice Erfahrungen gestaltete und somit für die Nutzung durch blinde Nutzer kompatible, zunächst fiktive Oberfläche vorausgesetzt.

## 4.1 Atomare Aufgaben in HTML-basierten Oberflächen

HTML-basierte Oberflächen von Serverapplikationen weisen besondere Merkmale auf, die es ermöglichen, einen ‚atomaren‘ Aufgabensatz zu definieren. Dabei ist es unerheblich, um welche Art der Applikation es sich handelt. Web Content Management Systeme wie auch Webmailer nutzen identische HTML-Funktionalitäten. Für die Betrachtung im Rahmen dieses Beitrags werden ‚reine‘ HTML-Oberflächen vorausgesetzt, die nicht durch Java-Applets oder andere PlugIns bereichert werden. Dem Ziel, eine HTML-Oberfläche zu benutzen, liegt zunächst die Auswahlregel ‚benutze HTML-Oberfläche‘ (s. Abbildung 1) zugrunde, die eines der folgenden Teilziele auswählt:

- *Link finden und klicken (Navigation)* – ‚finde\_und\_klicke Link‘. Serverapplikationen mit HTML-basierten Oberflächen bieten dem Nutzer verschiedene Funktionen auf unterschiedlichen Seiten an. Bei einem Webmailer z.B. findet sich der Posteingangs-Ordner auf einer anderen HTML-Seite als die Darstellung einer Mail. Der Wechsel auf eine andere Seite erfolgt durch das Klicken eines Links, der vorher durch den Nutzer erkannt werden muss.
- *Formularfeld finden und ausfüllen* – ‚füll\_aus Formular‘. Um Daten an eine Serverapplikation übergeben zu können, ist es notwendig die entsprechenden Informationen von einem Browser an den Server zu schicken. HTML sieht dafür Formulare vor, in die der Nutzer Daten eingibt. Die eingegebenen Daten werden durch das Absenden des Formulars an den Server geschickt.
- *Seiteninformationen finden und auswerten* – ‚finde\_und\_lies Inhalt‘. Informationen, die dem Nutzer Hilfestellung zur Nutzung geben und nicht als Link oder im Rahmen eines Formulars vermittelt werden können, müssen als gewöhnliche Inhalte in der Oberfläche hinterlegt werden. Es ist hinsichtlich der barrierefreien Umsetzung anzuraten, komplexe Seitenaufteilungen in sinnvolle Abschnitte zu gliedern und logische Strukturen durch entsprechende HTML-Konstrukte (Überschriften, Listen o.ä.) und passende Texte zu hinterlegen.

## 4.2 GOMS-Modellierung

Ziel der in diesem Beitrag beschriebenen Arbeit ist es zunächst, anhand der in 4.1 beschriebenen Aufgaben Nutzungsprinzipien herauszustellen, die im Rahmen einer Nutzerevaluation überprüft werden. Konkrete Aufgaben für die Berechnung von Ausführungszeiten werden erst später verfügbar sein. Für die Darstellung der vorgestellten Nutzungsstrategien wird eine dem GOMS-Dialekt NGOMSL (Natural GOMS Language, Kieras 1997) ähnliche Form gewählt. NGOMSL erlaubt das Modularisieren des Nutzungsmodells und legt eine formale Schreibweise für Ziele, Methoden, Operatoren und Auswahlregeln fest.

Das in Abbildung 1 dargestellte Modell verwendet sechs Operatoren:

- *rufe\_ab\_WM* (**mental operator**). Der Nutzer muss eine Information aus seinem Kurzzeitgedächtnis (Working memory) abrufen

- *tippe* (type a sequence operator). Der Nutzer gibt eine Folge von Zeichen über die Tastatur ein
- *warte\_auf* (wait operator). Der Nutzer wartet entweder die Ausgabe (Sprache oder Braille) des Systems oder die Zeit, die eine Seite für die Darstellung im Browser benötigt ab
- *?* (*entscheide*) (mental operator). Der Nutzer muss entscheiden, ob der aktuelle Status seinen Vorstellungen entspricht.
- *aktiviere* (keypress operator). Der Nutzer drückt eine Taste und aktiviert so einen Link oder ruft eine Funktion per Tastaturbefehl auf.
- *springe\_nächstes* (keypress operator). Der Nutzer weist den Screenreader an, auf das nächste Element in der Oberfläche zu wechseln.

<b>s:</b> benutze HTML-Oberfläche ? Navigieren: finde und klicke Link ? Daten eingeben: füll aus Formular ? Lesen: finde und lies Inhalt	<b>s:</b> finde_Element Element ? eindeutig, bekannt: finde_durch_Searching Element ? zweideutig, kein Listenmodus: finde_durch_Browsing Element ? zweideutig, Listenmodus: finde_durch_Liste Element	
<b>s:</b> ergänze_Eingabe Formularfeld ? Texteingabe: gib ein <Text> ? Option: o aktiviere Option <b>K</b> ? Auswahl: wähle aus Wert	<b>s:</b> schicke_ab Element ? Formular nicht komplett: teste Formular ? Formular komplett oder Link: eingabe Enter	
<b>m:</b> füll aus Formular finde_Element Formular (1) finde_Element Formularfeld ergänze_Eingabe Formularfeld o ? weitere Daten?: => (1) <b>M</b> schicke_ab Formular	<b>m:</b> gib ein <Text> o rufe_ab_WM <Text> <b>M</b> o tippe <Text> <b>T(n)</b> <b>m:</b> finde und klicke Link finde_Element Link schicke_ab Link	<b>m:</b> finde_durch_Searching Element wechsele_Modus Suche <b>M</b> o rufe_ab_WM Suchtext gib ein <Suchtext> o (2) aktiviere Suche <b>K</b> o warte_auf Fundstelle <b>W(t<sub>read</sub>)</b> o ? falsche Stelle: => (2) <b>M</b>
<b>m:</b> wähle aus Wert (6) finde Element nächster_Vorgabewert o ? unpassender Wert: => (6) <b>M(K)</b> o aktiviere Wert <b>K</b>	<b>m:</b> eingabe Enter o aktiviere Enter <b>K</b> prüfe Daten_bestätigt	<b>m:</b> prüfe aktion o warte_auf Seite <b>W(t<sub>request</sub>)</b> o warte_auf Seitenintro <b>W(t<sub>read</sub>)</b> o ? Aktion bestätigt <b>M</b>
<b>m:</b> finde_durch_Liste Element wechsele_Modus Liste o (4) springe_nächstes Element <b>K</b> o warte_auf Elementinhalt <b>W(t<sub>read</sub>)</b> o ? falsche Stelle: => (4) <b>M</b>	<b>m:</b> wechsele_Modus Modus o rufe_ab_WM Befehl <b>M</b> o rufe Modus <b>K</b> o warte_auf Feedback <b>W(t<sub>read</sub>)</b>	<b>m:</b> finde und lies Inhalt finde_Element Überschrift finde_Element nächstes_Element
<b>m:</b> finde_durch_Browsing Element o (3) springe_nächstes Element <b>K</b> o warte_auf Elementinhalt <b>W(t<sub>read</sub>)</b> o ? Sprunglink: aktiviere Link, => (3) <b>M(K)</b> o ? falsche Stelle => (3) <b>M</b>		<b>m:</b> teste Formular o (5) springe_nächstes Formularfeld <b>K</b> o ? Eingabe fehlt: ergänze_eingabe Feld <b>M</b> o ? kein submit-Button: => (5) <b>M</b> eingabe Enter

Abbildung 1 Dies ist die leicht verkürzte Darstellung des entwickelten GOMS-Modells. Der Startpunkt zur Erreichung des Ziels ‚benutze HTML-Oberfläche‘ ist grau hinterlegt. Jeder Block steht für eine Auswahlregel (‚s‘) oder eine Methode (‚m‘) des GOMS-Modells. Operatoren werden durch ‚o‘ gekennzeichnet und sind am Ende einer Zeile jeweils einem der Basisoperatoren M, T, W und K zugeordnet. Die Ziele sind in einer der Kombination <Verb> <Substantiv> ähnlichen Form beschrieben (z.B. warte\_auf Seite.) Das Verb zeigt entweder eine Auswahlregel, eine Methode oder einen Operator an. Ziffern in Klammern zeigen Sprungmarken innerhalb einer Methode an.

### 4.3 Ergebnisse des GOMS-Modells

Das vorgestellte Modell der grundlegenden Nutzungsstrategien zeigt in einigen Punkten Auffälligkeiten, die bei der Betrachtung des Nutzerverhaltens blinder Nutzer zu berücksichtigen sind.

- *Anzahl der Operatoren.* Im Vergleich zu NGOMSL-Modellen visuell wahrnehmbarer Oberflächen benötigt das vorgelegte Modell deutlich weniger Operatoren (hier 6 statt 14 in Kieras 1997). Bei Kieras hängen 7 Operatoren unmittelbar mit visuellen Interaktionen zusammen. Im vorgelegten Modell erfordern die Tastenkommandos das Erinnern der korrekten Syntax. Dies erfordert die vermehrte Nutzung von mentalen Operatoren.
- *Wartezeit.* Während Wartezeiten bei GOMS-Analysen visueller Interaktionen in der Regel vernachlässigt werden, zeigt das vorgestellte Modell, dass blinde Nutzer in regelmäßiger Folge auf Ausgaben des Systems warten müssen, die über das Gehör oder den Tastsinn erfasst werden.
- *$W(t_{read})$ .* Die Zeit, die der Screenreader zur Ausgabe eines Inhalts benötigt, ist abhängig von der Ausgabegeschwindigkeit des Sprachsynthesizers / der Braillezeile und der Länge des vorzulesenden Textes.
- *Abbruch in  $W(t_{read})$ .* Nutzer von Screenreader können schon während des Vorlesens eines Elements entscheiden, zum nächsten Element zu springen. Somit kann erwartet werden, dass die Wortlänge (bzw. Silbenanzahl) als grundlegendes Maß ungeeignet ist.
- *$W(t_{request})$ .* Die Zeit, die der Nutzer beim Laden einer neuen Seite warten muss, bis die Wahrnehmung beginnen kann. Diese Ausprägung des Wait-Operators betrifft nicht nur blinde Nutzer.
- *Listenmodus.* Die Nutzung von durch den Screenreader erzeugten Listen gleichartiger Elemente erfordert mehr Nutzeraktionen als das Springen von Element zu Element. Zudem erlauben es Listen nicht, Sprungmarken, die in die Oberfläche eingebettet sind, zu nutzen. Nutzerberichten ist zu entnehmen, dass die Listen-Funktion (gerade für das Auffinden von Links) dennoch genutzt wird. Gründe hierfür sind durch entsprechende Interviews festzustellen.

## 5 Fazit

Das Modell zeigt, dass GOMS auf die Mensch-Computer-Interaktion blinder Nutzer gut anwendbar ist. Durch die entwickelten Nutzungsstrategien konnten Unterschiede zur visuell basierten Interaktion aufgezeigt werden. Bereits die Strategien (ohne dass die GOMS-Analyse mit der Berechnung einer Ausführungsdauer abgeschlossen wurde) bestätigen Designempfehlungen hinsichtlich barrierefreier Gestaltung. So ist es sinnvoll, Feedback-Informationen an einem sehr frühen Zeitpunkt in der Seite anzubieten, um die Orientierungsphase für blinde Nutzer mit entsprechenden Inhalten anzureichern. Aufgrund der hohen Anzahl von Wartezeiten scheint es zudem sinnvoll, Oberflächen so zu reduzieren, dass ein blinder Nutzer nur wenige Seitenelemente zur Ausführung einer Aufgabe berücksichtigen muss. Nutzertests, die Mitte 2005 folgen, werden zur Verifikation des Modells und zur Ermittlung der Ausführungszeiten für die Basisoperatoren genutzt. Unter Umständen ist es notwendig, die Auswahlregeln anzupassen. Weiterhin ist zu klären, ob sich die Ausführungsdauer auch auf andere Nutzergruppen übertragen lässt.

### Literaturverzeichnis

- Bundesministerium für Gesundheit und Soziale Sicherung (BMGS, 2002): Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (BITV). Verordnung und Anlagen: [http://www.bmgs.bund.de/deu/gra/gesetze/ges\\_4.cfm](http://www.bmgs.bund.de/deu/gra/gesetze/ges_4.cfm) (Stand 18.02.2005)
- Card, S. K.; Moran, T. P.; Newell, A. (1983): The psychology of Human-Computer Interaction. Hillsdale, New Jersey: Lawrence Erlbaum Associates
- Chisholm, W.; Vanderheiden, G.; Jacobs, I. (1999): Web Content Accessibility Guidelines 1.0 (W3C Recommendation 5-May-1999). <http://www.w3.org/TR/WCAG10/> (Stand 18.02.2005)
- Hellbusch, J.E. (2004): Barrierefreies Webdesign. Heidelberg: d-punkt.verlag
- John, B. E.; Gray, W. D. (1995): CPM-GOMS: An Analysis Method for Tasks with Parallel Activities. In: Conference companion on Human factors in computing systems, S. 393-394, New York, NY, USA: ACM Press
- John, B. E.; Kieras, D. E. (1996): The GOMS family of user interface analysis techniques: comparison and contrast. In: ACM Transactions on Human-Computer Interaction, Vol. 3, Issue 4, S. 320-351, New York, NY, USA: ACM Press
- Kieras, D. (1997): A Guide to GOMS Model Usability Evaluation using NGOMSL. In: Helander, M.; Landauer, T.K.; Prabhu, P. (Hrsg.): Handbook of Human-Computer Interaction, S. 733-766, Amsterdam: Elsevier
- Kieras, D.E. (2001): Using the Keystroke-Level Model to Estimate Execution Times. Online Handout: <ftp://www.eecs.umich.edu/people/kieras/GOMS/KLM.pdf> (Stand 21.02.2005)
- Robertson, V.; Shelat, B.; Stewart, T.; Travis, D.; Tynan, A. (2003): Accessibility Study of BBCi – Problems faced by users with disabilities. [http://www.bbc.co.uk/commissioning/newmedia/pdf/BBCi\\_Accessibility\\_Study\\_7-10-02.pdf](http://www.bbc.co.uk/commissioning/newmedia/pdf/BBCi_Accessibility_Study_7-10-02.pdf) (Stand 18.02.2005)
- Treviranus, J.; McCathieNevile, C.; Jacobs, I.; Richards, J. (2000): Authoring Tool Accessibility Guidelines 1.0 (W3C Recommendation 3 February 2000). <http://www.w3.org/TR/ATAG10/> (Stand 18.02.2005)

### Kontaktinformation

Henrik Eichstädt  
Technische Universität Ilmenau  
Institut für Medien- und Kommunikationswissenschaft  
Fachgebiet Multimediale Anwendungen  
Am Eichicht 1  
98693 Ilmenau  
  
henrik.eichstaedt@tu-ilmenau.de  
<http://www.tu-ilmenau.de/site/ifmk/heichstaedt.html>  
Telefon +49-3677-69 47 40 / Telefax +49-3677-69 47 24

### Danksagung

Den Reviewern vielen Dank für die hilfreichen Anmerkungen.