

Direkte Manipulation von akustischen Objekten durch blinde Rechnerbenutzer

Ludger Bölke und Peter Gorny
Carl von Ossietzky Universität Oldenburg

Zusammenfassung:

Es gibt verschiedene Systeme, die es sehgeschädigten Benutzern gestatten, mit akustischen Repräsentationen von grafischen Benutzungsschnittstellen zu arbeiten (Graphical User Interface - GUI). Diese Anpassungen von GUIs an die Bedürfnisse von Blinden bieten jedoch kein hinreichendes Feedback, um ihnen eine effiziente Benutzung der Maus zur Cursor-Steuerung zu ermöglichen.

In diesem Beitrag werden die Möglichkeiten zur direkten Manipulation von akustischen Objekten, den Hearcons, beschrieben. Dazu wird ein für die Steuerung der Mausbewegungen unabdingbares akustisches Feedback erzeugt, das die Auswahl und Bearbeitung hörbarer Objekte erlaubt. Die Grundlage für diese Ohr-Hand-Koordination ist eine Kombination von permanent klingenden aktiven Hearcons und einem akustischen Feedback, das die relative Position des Maus-Cursors zu einem Zielobjekt liefert. Durch die Ausnutzung des räumlichen Hörvermögens können die Benutzer auf der technischen Basis der Kunstkopf-Stereophonie die Hearcons nach eigenem Bedarf im Hörraum positionieren.

Über den geschilderten adaptiven Ansatz hinausgehend wird vorgeschlagen, für Sehbehinderte spezielle Benutzungsschnittstellen auf der Basis der skizzierten Technik zu entwickeln. Das Konzept für das "assistive user interface" SPUI-B (StereoPhonic User Interface for Blind) wird vorgestellt.

Schlüsselworte: Akustisches Feedback, Hearcon, Direkte Manipulation, Blinde Benutzer, Ohr-Hand-Koordination, Akustische Benutzungsschnittstelle

1. Einleitung

Blinde Menschen müssen ihre Sehschädigung durch taktile und akustische Wahrnehmung kompensieren. Um ihnen den Zugang zum Rechner zu ermöglichen, sind die Sprachausgabe (vor allem in den USA) und die Braille-Ausgabe (vor allem in Deutschland) entwickelt worden. Diese adaptiven Techniken basieren auf dem sogenannten Screenreader, der die Ausgaben der Anwendungsprogramme direkt dem Bildschirmspeicher entnehmen. In den textbasierten Systemen können die auf dem Bildschirm ausgegebenen Zeichen im ASCII-Code in einem Textpuffer abgelegt und anschließend direkt in Braille-Zeichen oder Sprache transformiert werden. In grafischen Systemen wird die Bildschirmausgabe dagegen als Bitmap gespeichert. Ein Bit enthält nun die Information, ob ein Pixel auf dem Bildschirm gesetzt wird oder nicht. Das aus dieser Bitmap entstehende Pixelmuster wird dann vom Betrachter als Text oder Grafik wahrgenommen, der nicht direkt in taktile oder akustische Form umgewandelt werden kann.

Die grafischen Systeme bilden die technische Basis für grafische Benutzungsoberflächen (GUIs), die eine intuitive Benutzung ermöglichen sollen. Objekte und

Funktionen werden in textueller oder bildlicher Form direkt auf dem Bildschirm dargestellt; sie werden referenziert, indem sie mit einem Mausklick selektiert oder mit der Maus bewegt werden (drag and drop). Um dem Pixelmuster die textuelle Information zu entnehmen, sind spezielle zeit- und rechenintensive Mustererkennungsprogramme nötig, die jedoch (noch) nicht beliebige Zeichensätze erkennen können. Dies war lange Zeit mit ein Grund dafür, daß Blinde von der Benutzung grafischer Oberflächen ausgeschlossen waren.

Eine Lösung dieses Problems wird durch das "off-screen-model" gegeben [16]. Es fängt alle Funktionen ab, die sich auf die Bildschirmausgabe beziehen, und speichert die Informationen in ASCII-Format in einer Datenbank. Um den blinden Benutzern die Arbeit mit einer grafischen Oberfläche zu ermöglichen, werden außerdem alle Cursor-Bewegungen verfolgt und die textuelle Information über das Objekt, welches gerade vom Cursor berührt wird, an die Sprachausgabe oder die Braille-Zeile weitergegeben.

Das folgende Kapitel beschreibt drei spezielle Probleme, mit denen blinde Benutzer konfrontiert sind, wenn sie mit grafischen Benutzungsoberflächen arbeiten. Dabei wird deutlich werden, daß sie mit den gegebenen Adaptionen von Anwendungsprogrammen nicht ebenso effektiv und effizient arbeiten können wie sehende Benutzer. Im Anschluß daran wird die direkte Manipulation von akustischen Objekten erläutert. Diese Technik ermöglicht blinden Benutzern eine effiziente direkte Manipulation von Objekten mit der Maus und sie bildet die Basis für das "assistive" System SPUI-B (StereoPhonic User Interface for Blind).

2. Grafische Benutzungsoberflächen und blinde Benutzer

2.1 GUIs und ihre Barrieren für blinde Benutzer

Die Vorteile von GUIs (gegenüber textuellen Schnittstellen mit kommando- und menüorientierter Steuerung) sind bereits auf breiter Ebene in den letzten Jahren diskutiert worden. Deshalb werden hier nur die vier, im Zusammenhang mit diesem Thema wichtigsten Vorteile genannt:

1. Die Metapher gewohnter Umgebungselemente, die mit Icons grafisch dargestellt werden; diese Icons sind Referenzen auf Datenstrukturen, Programme, Funktionen usw.
2. Die direkte Manipulation der sichtbaren Objekte mit unmittelbarem visuellen Feedback für alle Aktionen.
3. Die einfache Steuerung parallel zugänglicher bzw. ablaufender Programme und Funktionen.
4. Die topographische Strukturierung und Organisation der sichtbaren Objekte auf dem Bildschirm durch Zuordnen von Semantik zu den Positionen.

So wie GUIs auf der einen Seite weniger geübten Benutzern die Arbeit am Rechner erleichtern, so erschweren sie auf der anderen Seite sehgeschädigten Benutzern die Arbeit und verursachen sogar neue Hindernisse für sie.

Boyd et al. [2] geben folgende drei Problembereiche an, die die Ursache dafür bilden, daß blinde Benutzer nicht ebenso effektiv und effizient mit GUIs arbeiten können wie Sehende:

1. Die *Pixel-Barriere* bezieht sich auf das Problem, daß die Bildschirmausgaben im Pixelformat abgespeichert werden, welches von einem gewöhnlichen Screenreader nicht gelesen werden kann. Eine Lösung für dieses *technische Problem* ist das bereits erwähnte *off-screen-model*.
2. Die *Maus-Barriere* zielt auf das Problem, daß blinde Benutzer die Maus nicht effektiv als Eingabegerät nutzen können. Dieses *motorische Problem* resultiert daraus, daß ein geeignetes Feedback zur Maussteuerung fehlt. Sehende sind aufgrund der *Auge-Hand-Koordination* in der Lage, die Maus zu benutzen. Sie *sehen* Cursor und Zielobjekt und sie *sehen*, wie sich die relative Lage des Maus-Cursors zu den Objekten verändert, wenn die Maus bewegt wird. Blinde erhalten keine Rückmeldung über die relative Lage des Cursors zu den Objekten und von daher können sie GUIs nicht in derselben Art und Weise benutzen wie Sehende.
3. Die *Grafik-Barriere* stellt das schwierigste Problem dar, da es ein *semantisches Problem* ist. Mehr und mehr Informationen werden grafisch präsentiert; zudem geben Topographie und Topologie der Objekte zusätzliche Hinweise über die Objekte selbst und ihre Beziehungen zueinander. Eine Transformation dieser grafischen Darstellungen in Ton- und Sprachausgabe ist immer mit einem Verlust von Informationen verbunden. Textuelle Beschreibungen von Grafiken sind häufig sehr lang, ungenau, kompliziert und umständlich ("Ein Bild sagt mehr als tausend Worte"). Eine technische Lösung des Problems ist das zweidimensionale taktile Display ("Stuttgarter Stiftplatte"); die Nachteile sind jedoch die geringe Auflösung (8 dpi) und die hohen Anschaffungskosten.

Wie bereits erwähnt, ist die Grafik-Barriere ein semantisches Problem und es ist fraglich, ob es je zufriedenstellend gelöst werden kann. Grafische Symbole werden verwendet, um ein Objekt in offensichtlicher Weise zu repräsentieren. Icons z.B. sind häufig vereinfachte Bilder von Objekten des alltäglichen Lebens, z.B. Mülleimer oder Radiergummi. Werden solche Icons in eine taktile Ausgabe mit einer wesentlich geringeren Auflösung als der Bildschirm transformiert, so ist es sehr zweifelhaft, ob blinde Benutzer die Objekte ebenso leicht erkennen können, da sie die Objekte nur vom Ertasten in Originalgröße kennen. Die verkleinerte taktile Figur wird nicht eindeutig zu erkennen sein und sie verliert vollständig den metaphorischen Bezug zur realen Umgebung.

Blinde Benutzer müssen also die Bedeutung der taktilen Icons als abstrakte Darstellungen erlernen und dieser Lernprozeß steht in krassem Widerspruch zu dem Ziel der GUIs, intuitiv und offensichtlich benutzbar zu sein. Dieser Nachteil kann auch nicht durch die Entwicklung kostengünstigerer zweidimensionaler taktiler Displays mit einer höheren Auflösung als die zur Zeit erhältlichen Displays beseitigt werden [8]. Nichtsdestotrotz ist eine Weiterentwicklung solcher Displays wichtig und wird sicherlich die Möglichkeiten der interaktiven Arbeit mit schematischen Darstellungen von Objekten erweitern.

Es gibt einige Projekte, die versuchen, Grafiken in akustische Ausgaben zu transformieren [17], oder die Bilder für Blinde beschreiben [12], aber es existieren zur

Zeit noch keine praktischen Lösungen, um beliebige Grafiken akustisch leichtverständlich darzustellen.

2.2 Adaptionen existierender GUIs für blinde Benutzer

Zur Zeit existierende Adaptionen überwinden nicht die drei eben aufgeführten Barrieren. Im allgemeinen lösen sie die Pixel-Barriere, indem sie das off-screen-model verwenden und visuelle Objekte auf akustische Objekte im dreidimensionalen Hörraum abbilden. Häufig können blinde Benutzer zum Navigieren auf der Benutzungsoberfläche nur die Cursor-Tasten verwenden.

Im *GUIB*-Projekt (Textual and Graphical User Interfaces for Blind People) [12] wird z.B. der Bildschirminhalt von *MS-Windows* zum einen in eine taktile Form übersetzt, wobei die Topographie der Objekte beibehalten wird; zum anderen wird die taktile Ausgabe um die akustische Ausgabe mit Raumklang erweitert. Jedem Objekt der GUI - auch dem Cursor - ist ein Geräusch zugeordnet. Die Geräuschquellen der Objekte erscheinen im Hörraum an den gleichen Positionen wie die sichtbaren Objekte auf dem Bildschirm. Bewegt sich der Cursor auf dem Bildschirm, so bewegt sich die Geräuschquelle in derselben Richtung im akustischen Raum. Auf diese Art erhält der Benutzer die Information über die absolute Position des Cursors auf dem Bildschirm; die übrigen Objekte ertönen erst, wenn sie vom Cursor berührt werden. Mit diesem akustischen Feedback ist jedoch ein direktes Selektieren eines Objektes nicht möglich, da kein Feedback über die relative Lage des Cursors zum Zielobjekt gegeben wird. Diese Situation ist für Sehende vergleichbar mit einem schwarzen Bildschirm, auf dem nur der Cursor zu sehen ist und die Objekte erst aufleuchten, wenn sie vom Cursor berührt werden.

Der *SonicFinder* [9, 10, 11], eine Anpassung der Finder-Oberfläche des Macintosh für sehbehinderte Benutzer, arbeitet nach demselben Prinzip. Die Objekte sind als *akustisch passive* Objekte implementiert, die nur dann ertönen, wenn ein bestimmtes Ereignis eingetreten ist. In einer späteren Version des *SonicFinders* werden sogenannte *Soundholder* eingeführt, die an ein Objekt gebunden werden können und die akustisch Auskunft über ihre relative Position zum Cursor geben. Sie tönen permanent und ihre Lautstärke fällt mit steigender Entfernung zum Cursor, d.h. entfernt sich der Cursor von einem Objekt, so ertönt der diesem Objekt zugeordnete Soundholder leiser, und bei Annäherung wird er lauter. Da kein Raumklang verwendet wird, können die Soundholder nicht lokalisiert werden, um z.B. die Maus sofort in die richtige Richtung zu bewegen. Zudem bleibt das Grundproblem erhalten: wie finde ich ein Objekt, bevor ihm ein Soundholder zugeordnet ist?

Im *Soundtrack*-Projekt [6, 7] werden den absoluten Positionen der Objekte verschiedene Töne zugeordnet (Keyboard-Metapher). Tests ergaben jedoch, daß nur ein Musiker in der Lage war, den Tönen die entsprechende Positionsinformation zu entnehmen. Alle übrigen sehgeschädigten Testpersonen zählten die bereits vom Cursor passierten Objekte, um zum Zielobjekt zu gelangen.

Im *Mercator*-Projekt [13] wird blinden Benutzern der Zugang zu XWindows unter Unix ermöglicht. In diesem Projekt wird das Navigationsproblem dadurch gelöst,

daß die grafische Anordnung durch eine baumartige Hierarchie ersetzt wird. Um ein Objekt zu selektieren, muß die Baumstruktur mit Hilfe der Cursor-Tasten durchsucht werden. Die Objekte ertönen wiederum nur dann, wenn ein bestimmtes Ereignis eingetreten ist.

Zusammenfassend kann festgestellt werden, daß die bekannten Adaptionen grafischer Benutzungsoberflächen nicht alle drei Barrieren überwinden. Sie ermöglichen es Blinden nicht, GUIs in derselben Art und Weise wie die Sehenden mit der Maus zu benutzen. Mit Hilfe des off-screen-models wird das Pixel-Problem gelöst, jedoch ist die graphische Benutzungsoberfläche häufig auf eine textuelle menü-/kommandoorientierte Schnittstelle reduziert, deren Objekte mit den Cursor-Tasten selektiert werden können. Die topographische Anordnung der Objekte hat keinerlei Bedeutung (Ausnahme GUIB-Projekt); eine direkte Manipulation der Objekte ist nicht möglich, da ein ausreichendes akustisches Feedback nicht gegeben wird. Dieses Feedback muß direkt und nicht indirekt, wie etwa bei der Keyboard-Metapher, wahrnehmbar sein.

3. Direkte Manipulation akustischer Objekte

Die Basis für die akustische direkte Manipulation wird zum einen gebildet durch akustisch *aktive* Objekte, den *Hearcons*, die im 3-D-Raum positioniert werden, und zum anderen durch ein akustisches Feedback für die *Ohr-Hand-Koordination* zur Steuerung der Cursor-Bewegungen mit der 3-D-Maus oder anderen geeigneten 3-D-Zeigegeräten.

3.1 Hearcons

In unserem System soll jedes relevante Objekt der Benutzungsoberfläche durch ein sogenanntes Hearcon repräsentiert werden. Ein Hearcon ist charakterisiert durch:

- das Geräusch, welches das Objekt repräsentiert,
- seine Lautstärke,
- seine Positionskoordinaten im Raum,
- seine räumliche Ausdehnung.

Die verwendeten Geräusche werden dabei nicht auf bestimmte Geräuschklassen eingeschränkt, wie etwa bei den *auditory icons* in Gaver's *SonicFinder* [9], bei dem nur Umweltgeräusche verwendet werden, oder bei Blattner's *earcons* [1], die aus synthetischen Tonfolgen zusammengesetzt sind. Den Hearcons kann jedes Geräusch zugeordnet werden, welches zu dem gegebenen Objekt paßt. Die Koordinaten geben die aktuelle Position im Raum an; die räumliche Ausdehnung ist ein künstliches Attribut für eine Geräuschquelle, die eigentlich als punktförmig angenommen wird. Sie wird jedoch benötigt, um den Raum abzugrenzen, in dem ein Hearcon als vom Cursor selektiert gilt.

Im Unterschied zu bisherigen Adaptionssystemen werden Hearcons als akustisch aktive Objekte realisiert, die permanent klingen und nicht nur dann, wenn ein bestimmtes Ereignis eingetreten ist. Sie bilden die Grundlage für das akustische Feedback, um blinden Anwendern die effiziente Benutzung der Maus zu ermöglichen. Die Verwendung von Raumklang (Kunstkopf-Stereophonie) erlaubt ein beliebiges

Positionieren der Hearcons im Raum und die topographische Anordnung der Objekte kann dadurch wahrgenommen werden, daß alle Hearcons parallel klingen.

Weiterhin ist ein Hearcon als das *aktuelle* Hearcon ausgezeichnet. Nur die Ausgaben oder der Inhalt des Objekts, welches durch das aktuelle Hearcon repräsentiert wird, können vom Benutzer mit Hilfe eines Ausgabegerätes seiner Wahl (Sprachausgabe, Braille-Zeile, Bildschirm) wahrgenommen werden. Alle anderen Hearcons repräsentieren nur ihre Objekte. Ebenso beziehen sich alle Eingaben nur auf das Objekt des aktuellen Hearcons. Dies ist vergleichbar mit einem Fenster-System, bei dem alle Fenster, bis auf das aktuelle als Icons repräsentiert werden und nur der Inhalt des aktuellen Fensters auf dem Bildschirm dargestellt wird. Die Vorgehensweise bei der Eingabe verhindert nebenbei das 'unselected-window'-Problem, da der Benutzer nie im Zweifel darüber ist, welches Fenster gerade die Eingabe akzeptiert.

3.2 Das Selektieren eines Hearcons

Der Ansatz, die Hearcons als akustisch aktive Objekte zu realisieren, ermöglicht es dem Benutzer, ein Zielhearcon direkt zu selektieren, ohne den gesamten Raum danach zu durchsuchen. Der Benutzer hört die topographische Anordnung der Objekte und damit auch die Position seines Zielobjekts und er erkennt, in welche Richtung die Maus bewegt werden muß, um dorthin zu gelangen.

Um die Maus-Barriere zu überwinden, muß ein geeignetes Feedback zur Ohr-Hand-Koordination, die die Handbewegungen zum Positionieren des Cursors steuert, realisiert werden. Der Benutzer benötigt ein permanentes und unmittelbares Feedback, inwiefern sich die relative Position des Cursors gegenüber dem Zielobjekt verändert, wenn die Maus bewegt wird. Aus diesem Grund ändert sich die akustische Ausgabe aller Hearcons, wenn der Maus-Cursor bewegt wird.

Es werden im Projekt folgende Alternativen für das akustische Feedback untersucht:

1. Jede Veränderung der Position des stillen Cursors resultiert in einer Änderung der akustischen Parameter (z.B. Tonhöhe, Lautstärke, Timbre) der Hearcons. Beispielsweise erklingt ein Hearcon lauter oder im Ton klarer, wenn sich der Cursor nähert, und es klingt leiser oder im Ton dumpfer, wenn sich der Cursor entfernt. (Analogie: das Cursor-Icon ist unsichtbar, der Helligkeitsgrad der anderen Icons gibt die Nähe zum Cursor an.)
2. Dem Cursor selbst ist ein Hearcon zugeordnet und die Geräuschquelle bewegt sich mit dem Cursor im Raum. Alle übrigen Hearcons klingen unverändert weiter.
3. Eine Bewegung des klingenden Cursors führt dazu, daß sich die Geräuschquelle des Cursors im Hörraum entsprechend mitbewegt und daß sich die akustischen Parameter der Hearcons ändern (Kombination der Alternativen 1 + 2).
4. Der Cursor ist an einer festen Position und eine Bewegung der Maus resultiert in einer Bewegung aller Hearcons. (Analogie: der Benutzer bewegt sich zusammen mit dem Cursor durch eine Artificial Reality.)

Welche Alternative am besten geeignet ist, um die Ohr-Hand-Koordination zu steuern und welche am meisten das Selektieren eines Hearcons in einer gegebenen

Anwendung unterstützt, muß in Versuchen gezeigt werden, die im Projekt in enger Zusammenarbeit mit den Fachbereichen Physik (Akustik) und Psychologie im Rahmen des gemeinsamen Graduiertenkollegs Psychoakustik erfolgen.

Das Berühren eines Hearcons mit dem Cursor muß unmittelbar angezeigt werden. Das System *AudioWindows* [4, 5] benutzt dazu den sogenannten "spotlight". Die Idee hinter dem Spotlight ist der "gerade noch erkennbare Unterschied" (Just noticeable difference, kurz *jnd*), einem wichtigen Aspekt der Psychophysik: es ist die kleinste erkennbare Unterscheidung in der Wahrnehmung (akustisch, visuell, taktil) zwischen zwei Objekten [15]. Unterhalb dieses Schwellwertes werden nur ein Objekt oder zwei identische Objekte erkannt. Eine andere Lösung ist ein zusätzlicher Ton, der beim Berühren eines Hearcons ertönt.

Dasselbe Problem entsteht, wenn ein gefundenes Hearcon selektiert wird. Entweder ändern sich die akustischen Parameter des Hearcons ein wenig, oder aber ein weiterer Ton wird eingeführt, um die Selektierung des Hearcons anzuzeigen.

3.3 Die Manipulation der Hearcons

Im folgenden werden einige weitere Funktionen zur Manipulation der Hearcons kurz vorgestellt.

Verschieben eines Hearcons

Das Verschieben eines Hearcons erfolgt in der gleichen Art und Weise wie bei GUIs. Ein Hearcon muß gefunden und selektiert werden, bevor es zu seiner Zielposition gezogen werden kann. Während des Verschiebens muß ein unmittelbares Feedback darüber erfolgen, wo sich das Hearcon gerade befindet. Dazu bewegt sich die Geräuschquelle entsprechend der Cursor-Position.

Erzeugen eines Hearcons

Bevor ein Hearcon irgendwo plaziert werden kann, muß es natürlich erzeugt werden. Dazu ruft der Benutzer die Funktion 'erzeuge_hearcon' entweder durch Maus-Doppelklick auf, wenn sich der Cursor auf keinem Hearcon befindet, oder er drückt eine Funktionstaste. Die Position des neuen Hearcons kann entweder

- jeweils eine feste Position im Raum (z.B. direkt vor dem Benutzer) sein, oder
- die Position des Maus-Cursors, als der Vorgang gestartet worden ist.

Nach dem Erzeugen eines neuen Hearcons befindet sich der Cursor implizit auf diesem Hearcon, um dem Benutzer die Möglichkeit zu geben, das neue Hearcon unverzüglich an seine Zielposition zu bewegen, ohne es noch einmal selektieren zu müssen.

Löschen eines Hearcons

Um ein Hearcon zu löschen, muß es zunächst gefunden und selektiert werden, bevor die 'lösche_hearcon'-Funktion aufgerufen wird:

- Die erste Alternative ist, das Hearcon physikalisch zu einem Mülleimer-Hearcon zu ziehen, und
- die zweite Alternative ist, die Funktion über eine Funktionstaste aufzurufen.

Ein Verändern der Größe eines Hearcons ('shrinking' in GUIs) ist bei diesem Ansatz nicht von weiterem Interesse, da die Hearcons die Objekte nur repräsentieren und keine Informationen über den Inhalt des Objekts ausgeben. Das Vergrößern eines Hearcons hätte nur den Effekt, daß die Fläche zum Finden dieses Hearcons wächst.

4. Einschränkungen des Ansatzes: Parallel klingende Hearcons

Dieser Ansatz ist natürlich einigen Einschränkungen unterlegen und löst nicht alle aufgezählten Probleme der Blinden im Umgang mit GUIs. Ebenso wenig können gegebene GUIs derart adaptiert werden, daß jedem Objekt der Oberfläche ein aktives Hearcon zugeordnet wird.

Das menschliche Gehör kann nur eine um einige Größenordnungen kleinere Anzahl von Informationen gleichzeitig bewußt wahrnehmen wie die Augen. Als Konsequenz können nur viel weniger aktive Hearcons gleichzeitig ertönen als visuelle Icons gleichzeitig auf dem Bildschirm erscheinen können, wenn sie noch bewußt wahrgenommen werden sollen. Ertönen zu viele Hearcons, ist es sehr zweifelhaft, ob ein einzelnes Hearcon separat herausgefiltert und lokalisiert werden kann - trotz des sogenannten 'Cocktail-Party-Effekts' (die Fähigkeit, innerhalb eines 'Volksgemurmels' einen ganz bestimmten räumlichen Bereich akustisch wahrzunehmen).

Es werden im Projekt folgende Lösungen näher untersucht:

1. Nicht alle Hearcons sind aktive Objekte und klingen permanent, sondern nur die in der Nähe des Cursors (siehe Gaver's Lösung im SonicFinder, inkl. der Soundholder).
2. Homogene Hearcons werden so gruppiert, daß sie wie *ein* Hearcon klingen. Ein akustisches Gruppieren von Geräuschen kann mit ähnlichen Geräuschen (gleiches Timbre) oder mit Tönen, die in der Tonhöhe dicht beieinander liegen, realisiert werden [3].
3. Hearcons können beliebig vom Benutzer gruppiert und die Gruppen im Klang gedämpft ("in den Hintergrund geschickt") werden.

Baumartige Strukturen (siehe Mercator-Projekt) werden nicht weiter verfolgt, da sie die räumliche Metapher verlassen.

Der Nachteil der ersten Lösung ist, daß nicht mehr alle Hearcons direkt selektiert werden können, da der Benutzer sie nicht alle hört. Soll ein Hearcon selektiert werden, welches nicht klingt, so muß der Benutzer entweder die Position des Hearcons kennen, um die Maus in die richtige Richtung zu bewegen, oder im schlimmsten Fall muß der ganze Raum abgesucht werden, bis das Zielobjekt gefunden ist. Eine Vorbedingung für Lösung 2 ist, daß überhaupt ähnliche Hearcons existieren, die dann gruppiert werden können. Die dritte Lösung schließlich erlaubt dem Benutzer beliebig zu gruppieren und der Gruppierung eine Bedeutung beizumessen, allerdings ist dieser Vorgang dann ein zusätzlicher Arbeitsschritt.

Es wird im Projekt experimentell untersucht, wieviel Hearcons parallel klingen und vor allem bewußt wahrgenommen werden können und welche Lösung am besten geeignet ist, wenn viele Hearcons parallel klingen sollen.

Als Schlußfolgerung kann festgehalten werden, daß Hearcons und die erläuterte Ohr-Hand-Koordination nicht in jedem Fall zur Adaptierung gegebener GUIs geeignet sind, da ernsthafte Probleme auftauchen, wenn zu viele Hearcons gleichzeitig präsentiert werden müssen. Statt der Suche nach einer Adaption, die den blinden Benutzern alle Möglichkeiten und Vorteile von GUIs eröffnet, wird im nächsten Kapitel das "assistive interface" SPUI-B (StereoPhonic User Interface for Blind) vorgestellt, welches blinden Benutzern die erwähnten Vorteile der Fenster-Systeme und die akustische direkte Manipulation eröffnet.

5. SPUI-B - eine assistive Technik

Zuerst ist die Frage zu beantworten, warum eine neue Spezialumgebung (in diesem Sinne assistive Technik) nur für blinde Benutzer entwickelt werden soll, wo doch bekannt ist, daß Benutzer mit Behinderungen gerade an adaptiven Lösungen, d.h. an Adaptionen der Programme für nicht- (oder weniger) behinderte Benutzer interessiert sind. Die Antwort ist, daß Adaptionen gegebener GUIs immer mit einem Verlust an Informationen verbunden sind (wie z.B. dem Verlust der Information, die in der topographischen Anordnung der Objekte verborgen ist). Ein assistiver Ansatz eröffnet vielleicht eine größere Chance, eine Spezialumgebung für Blinde zu entwickeln, die ihnen Zugang zu den Informationen ermöglicht, die z.B. in der topographischen und/oder der topologischen Anordnung der Objekte enthalten ist, und ihnen dieselben funktionalen Vorteile bietet, wie die GUIs den Sehenden. Daneben soll es natürlich möglich sein, innerhalb der Spezialumgebung mit anderen Adaptionen graphischer Benutzungsoberflächen zu arbeiten, so daß blinde Benutzer trotzdem in der Lage sein werden, mit sehenden Kollegen an einem Bildschirm bzw. Arbeitsplatz zu arbeiten.

5.1 Ziele

Das Hauptziel von SPUI-B ist, blinden Benutzern die Vorteile der Fenster-Systeme zugänglich zu machen, ohne bloß gegebene GUIs zu adaptieren. Das ist erstens die Möglichkeit, Metaphern gewohnter Umgebungen zu realisieren, zweitens mehrere Anwendungen, die parallel laufen, auf einfache Art zu steuern, indem ohne großen Aufwand zwischen ihnen hin- und hergeschaltet werden kann, drittens die direkte Manipulation von Hearcons, um die Benutzungsoberfläche an die eigenen Bedürfnisse anzupassen und viertens mit diesen Mitteln der topographischen Anordnung der akustischen Objekte eine Bedeutung geben.

5.2 Entwurfskonzept

Die Basis von SPUI-B ist ein Fenster-System, welches den Benutzern erlaubt, eine beliebige Anzahl an Fenstern zu erzeugen, in denen beliebige Anwendungen gestartet werden können, genau wie in einem visuellen Fenster-System. Eines der Fenster ist das aktuelle Fenster, welches die Tastatureingabe akzeptiert und welches seine Ausgaben zu dem vom Benutzer gewählten Ausgabegerät schickt. Sehende Benutzer werden den Bildschirm bevorzugen, während blinde Benutzer die Braille-Zeile oder die Sprachausgabe wählen werden. In diesem Sinne ist SPUI-B nicht nur eine Spezialumgebung für Blinde, sondern es kann ebenso von Sehenden benutzt werden.

Jedes nicht-aktuelle Fenster wird durch ein Hearcon repräsentiert. SPUI-B erlaubt die Wahl beliebiger Geräusche, z.B. Musik oder Umweltgeräusche. Als Ausgabetechnik wird die Kunstkopf-Stereophonie verwendet. Die Bewegung der Maus wird durch die Ohr-Hand-Koordination gesteuert. So kann der Benutzer die Hearcons, wie weiter oben erläutert, direkt manipulieren. Die Hearcons können beliebig im Hörraum plazierte werden, so daß die topographische Anordnung für den blinden Benutzer eine Bedeutung erlangen kann, da er die Anordnung selbst gestaltet. Darüber hinaus kann er jedes Fenster direkt selektieren, da das zugehörige Hearcon lokalisiert werden kann. Als Resultat kann der Wechsel zwischen den Anwendungen ganz einfach und ohne großen Aufwand für die Blinden dadurch erfolgen, daß eines der nicht-aktuellen Hearcons angefahren und aktiviert wird.

Weiterhin dienen die Hearcons als Gedächtnisstütze, da der Benutzer hört, wenn eine Anwendung noch läuft. Um die Funktion als Gedächtnisstütze ebenso gut wie grafische Icons zu erfüllen, reicht es jedoch nicht aus, daß erkannt wird, daß noch *irgendeine* Anwendung läuft. Der Benutzer muß dazu auch noch erkennen können, *welche* Anwendung noch läuft.

Die Brauchbarkeit des Systems hängt hier in hohem Maße von der Wahl der Geräusche ab, die die entsprechenden Fenster und Anwendungen repräsentieren.

5.3 Die Wahl der Geräusche

Wie bereits erwähnt, ist SPUI-B nicht beschränkt auf eine spezielle Geräuschklasse. Falls es zu einem Objekt oder einer Anwendung ein entsprechendes Objekt im 'realen' Leben gibt, so wird das dazugehörige Geräusch zur Repräsentation des Rechnerobjekts herangezogen. Die Idee dieses Ansatzes ist, die diesen Geräuschen inhärente Information über das Objekt zu nutzen, so daß die Bedeutung dieses Hearcons nicht erst erlernt werden muß. Nicht alle Objekte der Rechnerwelt haben jedoch entsprechende hörbare Pendanten im realen Leben. In solchen Fällen muß ein Geräusch gewählt werden, das evtl. nicht in unmittelbarem Zusammenhang mit dem Objekt steht, und die Bedeutung dieses Hearcons muß erlernt werden.

Der Gebrauch von synthetischer Sprache erfolgt nur auf Wunsch des Benutzers, da zum einen die Hearcons akustisch aktive Objekte sind, die permanent klingen, und zum anderen Sprache ein recht langsames Ausgabemedium ist.

Mynatt faßt einige Anforderungen an die den Rechnerobjekten zugeordneten Geräusche zusammen [14]:

- Die konzeptuelle Abbildung: Werden die Objekte offensichtlich repräsentiert?
- Die Benutzerempfindung: Klingt das Geräusch angenehm?
- Die physikalischen Parameter: Keine gegenseitige Maskierung der Geräusche, gute Klangqualität,...

Zudem muß es möglich sein, zueinander in Beziehung stehende Objekte durch geringe Unterschiede in ihren akustischen Parametern zu repräsentieren, um die Relationen zu verdeutlichen.

Ausgehend von diesen Anforderungen folgt, daß Hearcons nur dann als Gedächtnisstütze fungieren können, wenn die konzeptuelle Abbildung von der Anwendung zur akustischen Darstellung erfolgreich ist.

Welche Geräusche den Forderungen am besten entsprechen, hängt sowohl von den entsprechenden Anwendungen als auch vom Benutzer selbst ab. Ein Musiker wird ein Hearcon sicherlich anders hören als ein Nicht-Musiker. Deshalb muß den Benutzern die Möglichkeit eingeräumt werden, sich geeignete Geräusche selbst auszusuchen.

5.4 Projektorganisation

Das Projekt begann 1993 und befindet sich noch in der Phase der Erforschung der grundsätzlichen Prinzipien. Seit Beginn 1995 ist es Bestandteil des Graduiertenkollegs Psychoakustik der Carl von Ossietzky Universität Oldenburg.

Als technische Plattform werden DOS-kompatible PCs verwendet, die mit Soundkarten und Braille-Zeile ausgestattet sind.

6. Zusammenfassung

Anhand der Diskussion um die Pixel-, Maus- und Grafik-Barriere wurde gezeigt, daß einige der Vorteile der grafischen Benutzungsoberflächen von Blinden nicht genutzt werden können. Die Grafik-Barriere ist das schwierigste Problem und gibt Anlaß zu der Vermutung, daß Blinde prinzipiell nicht in die Lage versetzt werden können, GUIs ebenso effektiv und effizient wie Sehende zu benutzen.

Gegebene Adaptionen von GUIs lösen das Problem der Pixel-Barriere, jedoch ist die Arbeit mit ihnen häufig auf das Finden und Selektieren von Funktionen und Objekten mit Hilfe der Cursor-Tasten reduziert. Die topographische Anordnung der Objekte ist für die Blinden kaum wahrnehmbar, so daß eine mögliche Informationsquelle verborgen bleibt. Eine akustische Lösung zur effizienten Benutzung der Maus wurde vorgestellt, die auf akustisch aktiven Hearcons und der Ohr-Hand-Koordination mit akustischem Feedback über die relative Position des Cursors zu den Hearcons beruht. Weitere Funktionen zur direkten Manipulation der Hearcons sind erläutert worden.

Zu erforschende Probleme des Ansatzes sind die Auswahl geeigneter Geräusche, um Objekte akustisch offensichtlich zu repräsentieren, und die Anzahl gleichzeitig klingender Hearcons.

Von den allgemeinen Betrachtungen wurde das Konzept einer Spezialumgebung abgeleitet, das zum Entwurf des StereoPhonic User Interface for Blind (SPUI-B) führt.

Literaturverzeichnis

- [1] M. M. Blattner, D. A. Sumikawa, R. M. Greenberg: *Earcons and Icons: Their Structure and Common Design Principles*, Human Computer Interaction 1989, Vol.4
- [2] L. H. Boyd, W. L. Boyd, G. C. Vanderheiden: *Graphics-Based Computers and the Blind: Riding the Tides of Change*, in: Proceedings of the 6th Annual Conference "Technology and Persons with Disabilities" Los Angeles, 20-23.3.1991
- [3] A. S. Bregman: *Auditory Scene Analysis - The Perceptual Organization of Sound*, MIT Press, Cambridge 1990
- [4] M. Cohen, L. Ludwig: *Multidimensional audio window management*, International Journal of Man-Machine Studies, 1991, Vol 34, pp.319-336
- [5] M. Cohen: *Throwing, pitching and catching sound: audio windowing models and modes*, International Journal on Man-Machine-Studies, 1993, Vol 39, pp 269-304
- [6] A. D. N. Edwards: *Design of Auditory Interfaces for Visually Disabled Users*, in *Human Factors in Computing Systems* Proceedings of the CHI '88, ACM SIGCHI
- [7] A. D. N. Edwards: *Soundtrack: An Auditory Interface for Blind Users*, Human Computer Interaction 1989, Vol.4
- [8] J. Fricke, H. Bähring: *A graphic input/output tablet for blind computer users*, in W. Zagler (Ed.): *Computers for handicapped Persons*, Proceedings of the 3rd International Conference, Vienna, July 7-9, 1992
- [9] W. W. Gaver: *The SonicFinder: An Interface that uses Auditory Icons*, Human Computer Interaction 1989, Vol.4
- [10] W. W. Gaver, R. B. Smith: *Auditory Icons in Large-Scale-Collaborative Environments*, in H. Diaper et al. (Eds.): *INTERACT '90*, Proceedings of the 13 3rd Int. Conference on Human-Computer Interaction, Cambridge, U.K.. 27-31 August, 1990
- [11] W. W. Gaver, R. B. Smith, T. O'Shea: *Effective Sounds in complex Systems: The Arkola Simulation*, in 'Reaching through technology', CHI '91 Proceedings, 1991
- [12] P. L. Emiliani (Project Manager): *Publications from the TIDE project GUIB until to September 1993*, GUIB Consortium, 1993
- [13] E. D. Mynatt, W. K. Edwards: *Mapping GUIs to Auditory Interfaces*, in Proceedings of the UIST'92, Nov.15-18, 1992, Monterey, CA
- [14] E. D. Mynatt: *Designing with Auditory Icons: How Well Do We Identify Auditory Cues*, in: Proceedings of the CHI'94 Conference, Boston, M., April 24-26, 1994
- [15] J. G. Roederer: *Physikalische und psychoakustische Grundlagen der Musik*, Springer Verlag Berlin Heidelberg New York, 1977
- [16] R. S. Schwerdtfeger: *Making the GUI Talk*, BYTE, December 1991

- [17] S. Smith, R. D. Bergeron, G. G. Grinstein: *Stereophonic and surface sound generation for exploratory Data Analysis* in CHI '90 Proceedings April 1990

Die Autoren bedanken sich bei den Mitgliedern des Graduiertenkollegs Psychoakustik der Universität Oldenburg, besonders bei Prof. Dr. Volker Mellert und seiner Abteilung.

Ludger Bölke und Peter Gorny

Carl von Ossietzky Universität Oldenburg - Fachbereich Informatik, D-26111 Oldenburg
EMail: Ludger.Boelke@informatik.uni-oldenburg.de, Gorny@informatik.uni-oldenburg.de

