

**Abschlussbericht**  
**über das vom Bundesministerium für Wirtschaft und  
Technologie (BMWi) geförderte**  
**Projekt „HyperBraille“**

<b>ZE: Metec - Ingenieur AG</b> <b>Hasenbergstr.31</b> <b>70178 Stuttgart</b>	<b>Förderkennzeichen:</b> <b>01 MT07001A</b>
Projektleitung: Herr Uwe Grotz Tel.: 0711/66603-26 Fax.: 0711/66603-33 E-Mail: ug@metec-ag.de	Autoren: -Herr Volker Hantschel Leiter Softwareabteilung -Herr Uwe Grotz Vorstand

Vorhabenbezeichnung:

**Verbundprojekt**      HyperBraille: Entwicklung einer neuartigen zweihändigen Handhabungs- und Bedienoberfläche für Interaktive Stiftplatten zur taktilen Erschließung visueller graphischer Bildschirm-Informationen und graphischer Notationen für Blinde

**Teilvorhaben**      Hard- und Software für Stiftplatte

Laufzeit des Vorhabens :

**01.04.2007 bis 31.03.2011**

Berichtszeitraum:

**01.04.2007 bis 31.03.2011**

Abgabedatum:

**26.07.2011**

## Inhalt:

1. Einleitung und Problemstellung des Projektes „HyperBraille“ .....	3
1. Einleitung und Problemstellung des Projektes „HyperBraille“ .....	3
1.1 Gesamtziel des Projektes .....	3
1.2 Teilziele des Vorhabens .....	4
1.2.1 Hardware Entwicklung .....	4
1.2.2 Software Entwicklung .....	4
2. Wissenschaftlicher und technischer Stand an den angeknüpft wurde .....	6
3. Ergebnisse der Hardwareentwicklung .....	7
3.1 Geräteaufbau aus den überarbeiteten Antriebselementen aus dem Forschungsvorhaben „Interaktives Braille-Display“ .....	8
3.1.1 Beschreibung des überarbeiteten Antriebselementes .....	9
3.2 Geräteaufbau aus den neuen Antriebselementen .....	13
3.2.1 Beschreibung des neuen Antriebselementes .....	14
3.3 Zusammenfassung Hardwareentwicklung .....	21
4. Ergebnisse der Softwareentwicklung .....	23
4.1 Gesamtsoftware Hyperreader .....	23
4.1.1 Aufbau .....	23
4.1.2 Installation .....	24
4.1.3 Handhabung .....	26
4.2 Softwarekomponenten .....	27
4.2.1 Softwarekomponente Stiftplattenansteuerung – AP6 .....	29
4.2.2 AudioRenderer, Tastaturhook, etc. - AP10 .....	31
4.2.3 Tabellenverwaltung - Excel – AP15 .....	41
4.2.4 Softwarekomponente Signierung – AP17 .....	43
4.2.5 Kalenderverwaltung – Outlook - AP19 .....	45
4.3 Zusammenfassung der Softwareentwicklung .....	47
5. Notwendigkeit und Angemessenheit der geleisteten Arbeit .....	48
6. Fortschritt bei anderen Stellen .....	49
7. Anwendungspotentiale und Verwertung der Ergebnisse .....	50
7.1 Einsatzbereiche .....	50
7.1.1 Beruf, Schule, Studium .....	50
7.1.2 Öffentliche Gebäude, Museen, Institute .....	50
7.1.3 Forschungsprojekte .....	50
7.1.4 Privater Bereich .....	50
7.1.5 Sonstige Bereiche .....	51
7.2 Zielgruppen .....	51
7.2.1 Zielgruppe der blinden Nutzer, die 2011 bereits im Beruf stehen .....	51
7.2.2 Zielgruppe der zukünftigen blinden Nutzer, die bereits zum Abitur mit HyperBraille-System gearbeitet haben und über ein Studium mit HyperBraille- System in eine Berufsausübung mit HyperBraille-System „hineinwachsen“ .....	51
7.2.3 Zielgruppe aller blinden Menschen unabhängig von der Ausbildung oder dem Alter .....	52
7.2.4 Zielgruppe Forschungsprojekte .....	52
7.3 Verwertung .....	52
7.4 Gebrauchswert einer HyperBraille-Einheit .....	53
8. Abbildungsverzeichnis .....	54
9. Erfindungen/ Schutzrechte .....	56
10. Veröffentlichungen / Literatur .....	57

# **1. Einleitung und Problemstellung des Projektes „HyperBraille“**

Für blinde Menschen wächst mit zunehmender Verbreitung vielfältiger Computer-Anwendungen, die immer weiter über reine Sprachanwendungen hinausgehen, das dringende Bedürfnis, Anteil an diesen multimodalen modernen Kommunikationsformen nehmen zu können. Dabei geht es nicht nur primär um eine zeitgerechte Teilnahme an Aus- und Weiterbildungsmöglichkeiten, die zunehmend neben reiner Textausgabe graphische und strukturierte Informationserfassung voraussetzen, sondern auch um die selbständige, barrierefreie, berufliche Benutzung von interaktiven PC-Arbeitsplätzen.

Deren volle Funktionalität lässt sich jedoch heute nur durch die Anwendung entsprechender vorhandener Software erschließen, die – nicht nur wegen ihrer ausschließlichen Ausrichtung auf den dominierenden Bedarf für Normalsichtige – eine weit über reine Schriftzeichen hinausgehende, intelligente und rationelle Kommunikationsfähigkeit auch in Form gemischter grafischer und symbolischer Darstellung voraussetzt.

Beispiele dieser über Schriftzeichen hinausgehenden Informationen sind z. B. das Anzeigen eines Status im Ablauf eines modernen Programmsystems für Textverarbeitung oder Tabellenkalkulation, die Fensterbeschreibung in einer Windows-Anwendung oder eine rein grafische Darstellung. Ein weiterer Bedarf besteht in der großflächigen Darstellung entweder mehrerer Lesezeilen, oder gemischter Lese- und grafischer Informationen, z. B. zur Beschreibung bzw. Erklärung eines Grafen oder Diagramms.

## **1.1 Gesamtziel des Projektes**

Mit dem Entwicklungsprojekt HyperBraille soll ein neuartiges Handhabungs- und Bedienungssystem geschaffen werden, mit dem blinden und sehbehinderten Menschen mit Hilfe von interaktiven Stiftplatten ein autarker, barrierefreier Zugang zu wesentlichen textlichen und graphischen Informationen ermöglicht wird, die heute in elektronischer Form auf visuell genutzten Bildschirmen ausgegeben werden und in diesem Umfang Blinden und Sehbehinderten bisher weitgehend verschlossen bleiben.

Die Verwirklichung dieses Entwicklungsziels schafft damit alle Voraussetzungen, dass Sehbehinderte erstmals viel umfassender Zugang zu allen Kommunikationsmöglichkeiten erhalten, die für Normalsichtige heute üblicherweise über den Bildschirm eines PC-Systems zur Verfügung stehen.

Berufe bzw. Arbeitsplätze, die im Wesentlichen die Fähigkeit zur selbständigen, uneingeschränkten Kommunikation mit und über ein PC- System mit Bildschirmausgabe voraussetzen, können damit in Zukunft von Blinden und Sehbehinderten eigenständig und weitgehend vollwertig besetzt werden. Besonders wichtig ist auch die sich grundsätzlich mit dem System HyperBraille ergebende Möglichkeit, dass Blinde ihren Beruf – soweit Interesse besteht - auch weitgehend vom heimischen PC mit Internet- bzw. Modemanschluss ausüben können.

## **1.2 Teilziele des Vorhabens**

Das Vorhaben verfolgt zwei kohärente Teilziele:

### **1.2.1 Hardware Entwicklung**

Die interaktive Stiftplatte in der Größe ca. einer DIN A4 Seite mit 60 Zeilen mal 120 Spalten im Rasterabstand von 2,5 mm soll in Bezug auf Mobilität, d. h. insbesondere auf Energieverbrauch, Gewicht und Baugröße, optimiert werden. Die Baugröße und das Gewicht sind so weit zu reduzieren, dass die Stiftplatte transportiert werden kann. Hierzu sind die „Umweltfaktoren“ wie Schockbelastung, Temperatureinflüsse, Luftfeuchtigkeit etc. zu berücksichtigen, damit ein reibungsloser Betrieb der Stiftplatte im rauen Alltag möglich ist. Ziel ist es, dass dieses Arbeitsmittel (= Bildschirmsersatz) für die Blinden und Sehbehinderten zu jeder Zeit und umfassend zur Verfügung steht.

Die Herstellkosten der Stiftplatte sind durch geeignete konstruktive Maßnahmen und Verfahrensentwicklungen auf ein Maß zu senken, dass die Stiftplatte breite Anwendung finden kann und der gesteigerte Nutzen gegenüber einer „klassischen“ Braille-Zeile in einem gesunden Verhältnis zu den höheren Anschaffungskosten liegt. Damit soll sichergestellt werden, dass dieses neue Informations-Handhabungssystem schnell den Standard bei den Blindenhilfsmitteln bildet und eine weite Verbreitung findet. Diese Kostenreduktion ist durch Optimierung der Modulbauweise, der Elektronik und vor allem der Montage der Stiftplatte zu erreichen.

Konkret sollen im Verbundprojekt HyperBraille Untersuchungen zur kostengünstigen Serienherstellung von Braille-Modulen, mit denen die Displayaufbauten realisiert werden können, durchgeführt werden. Hierzu werden zunächst die aus dem Forschungsvorhaben „Interaktives Braille-Display“ erarbeiteten Ergebnisse verwendet und diese für eine Serienherstellung optimiert. Im zweiten Schritt sind dann neue optimierte Module zu konstruieren bzw. herzustellen, bei denen diese Ergebnisse umgesetzt werden. Die Arbeiten erfolgen in enger Zusammenarbeit mit der Hahn-Schickard-Gesellschaft-IMAT bzw. dem Institut für Zeitmesstechnik, Fein – und Mikrotechnik (IFZM) der Universität Stuttgart, die die Spritzgussteile metallisieren und laserstrukturieren sowie die elektronischen Bauteile bestücken. Das Institut für Mikroelektronik IMS-Chips in Stuttgart entwickelt für die Module die neuen NV- und HV-Chips zur Ansteuerung der Piezobiegeumwandler und zum Erfassen der Fingerposition.

### **1.2.2 Software Entwicklung**

Die Schaffung einer ausgereiften Software, die exakt auf die Belange der Blinden ausgerichtet ist und die Umsetzung der graphischen Informationen in taktil erfassbare Informationen bewerkstelligt, gilt als Grundvoraussetzung. Erst durch diese Kombination von Soft- und Hardware sind die Vorteile einer zweidimensionalen, flächigen Tastweise nutzbar. Notwendig zur sinnvollen Nutzung ist ein optimales Zusammenspiel zum einen zwischen der Software und der Stiftplatte, zum anderen aber auch zwischen der Basissoftware und der Anbindung an die Anwendungsprogramme. Hier ist das Ziel, die einzelnen Komponenten so zusammenzufügen, dass eine durchgängige Software entsteht, die immer nach den gleichen Mustern arbeitet.

Die HyperBraille-Software ist so anzupassen, dass alle Standardanwendungen wie Internet-Explorer, MS Word, MS Excel, MS Power Point etc. problemlos und einfach auf der Stiftplatte darstellbar und bedienbar sind. Dieses Zusammenspiel von Soft- und Hardware garantiert die Verbreitung und Akzeptanz des neuartigen Handhabungssystems und gibt die massiven Vorteile an den User weiter.

Konkret sind die Daten der Anwendungsprogramme MS Excel, MS Word, MS PowerPoint, Firefox, MS Outlook etc. so aufzubereiten, dass ein optimales Arbeiten und Verstehen auf der zweidimensionalen Stiftplatte möglich ist und die Zusatzinformationen schnell und reibungslos von den blinden und sehbehinderten Menschen aufgenommen werden können. Die Stiftplatte soll dabei zur Ein- und Ausgabe der Daten dienen und ähnlich einem Touchscreen arbeiten. Unterstützt wird dieses zudem durch Sprachausgabe.

## **2. Wissenschaftlicher und technischer Stand an den angeknüpft wurde**

Die Metec AG hat schon 1984 ein erstes elektromagnetisches Flächendisplay entwickelt. Von diesem Flächendisplay wurden 5 Geräte aufgebaut. Dieses Flächendisplay war noch ohne Touchsensorik und mit einem Punktabstand von 3 mm ausgeführt. Die Geräte dienten zu ersten Tests mit graphischen Darstellungen und führten über die Jahre zur Softwaregrundlage „TAWIS“. Mit dieser Software war es schon möglich, gewisse Darstellungen und Graphiken zu testen. Texte wurden wahlweise in Schwarzschrift und Brailleschrift dargestellt. Diagramme konnten mit Brailleschrift ergänzt bzw. die Achsenbeschriftung dargestellt werden. Das Programm zeigte die Möglichkeiten eines Flächendisplays auf. Die gewonnenen Ergebnisse und Erfahrungen sowie auch Limitationen bildeten die Grund- bzw. Ausgangslage bei der Entwicklung der HyperBraille-Software.

In Bezug auf die Hardware entstand in dem vom Bundesministerium für Forschung und Bildung geförderten Projekt Brailledis die Grundlage der HyperBraille-Geräte. In diesem Projekt wurde ein 10 Punkte Modul in MID Technik entwickelt. Die Module waren mit je einem Touchsensor ausgestattet, um Eingabebefehle positionsgenau an die Software zu übergeben. Die Module wurden entwickelt, um einen flächigen Aufbau mit Touchsensorik in Piezotechnologie zu verwirklichen. Im Projekt Brailledis wurde ein Demonstrator mit 96 Modulen und einer Fläche von 80 x 75 mm aufgebaut. Flächendisplays mit einer Matrix von 120 x 60 Punkten und damit 720 dieser Module entstanden erst im Projekt HyperBraille. Damit standen relativ früh Testgeräte für die Usertests und zur Softwareentwicklung zur Verfügung. Diese Module waren weltweit die einzig verfügbaren, um ein Flächendisplay mit einer Größe entsprechend etwa einer DIN A4 Seite mit Touchsensorik aufzubauen. Der technische Aufbau ist in Kapitel 3.1 näher erläutert.

### 3. Ergebnisse der Hardwareentwicklung

1. Mit dem Aufbau der ersten 8 Flächendisplays aus den Ergebnissen des Forschungsvorhabens „Interaktives Braille-Display“ konnten wichtige Erfahrungen in Richtung der Serienherstellung und Montage erworben werden.

Der Spritzgusskörper wurde in mehreren Schritten optimiert, um die Laserstrukturierung sicher zu stellen. Es zeigte sich, dass das Spritzgussteil in den konstruierten Abmessungen und in dem für das Verfahren notwendigen Material nur sehr schwer in konstanter Spritzgussqualität herstellbar ist. Das Einzelmodul ist durch das Laserstrukturieren in Verbindung mit den direkt auf den Spritzgusskörper gebondeten Chips sehr komplex. Kleinste Fehler in einem Bearbeitungsschritt führen zum Ausfall eines kompletten Moduls. In der Mehrzahl der Fälle ist das defekte Modul dann auch nicht mehr reparabel. Dadurch war die Ausfallquote schon beim Aufbau der Geräte für eine Serienherstellung deutlich zu hoch. Alle Optimierungen haben hier bei dieser Art von Modulen zu keinen, für eine Serienherstellung brauchbaren Ergebnissen, geführt. Die 8 Prototypgeräte konnten aber dennoch aufgebaut und betrieben werden. Damit konnten die Projektpartner die Software weiterentwickeln und Usertests gemacht werden. Hier zeigten sich recht schnell gewisse Limitierungen der Hardware. Zum einen waren die Module den Umweltbedingungen nicht gewachsen, so dass eine relativ hohe Anzahl innerhalb kurzer Zeit durch Stöße und Temperatureinflüsse versagten. Die Auflösung und Empfindlichkeit der Touchsensoren reichte für eine reibungslose Eingabe bzw. Gestensteuerung nicht aus. Aus diesen Erfahrungen und den Ergebnissen der Usertests wurden relativ schnell ein Pflichtenheft und eine Risikobewertung für das neue Display bzw. die neuen Module erarbeitet.

2. Mit den Anforderungen des Pflichtenheftes und der Risikobewertung wurde ein neues Modul konstruiert. Hier flossen sämtliche Erkenntnisse und Ergebnisse der ersten Generation von Geräten ein. Zudem wurde die Baugröße optimiert und auf eine kostengünstige Serienherstellung geachtet. Damit entstand ein komplett neues Flächendisplay, das in seiner Funktionalität und Herstellbarkeit deutlich verbessert wurde. Durch die Vorgaben der Anwender und das Mitwirken eines Designers wurde aus dem unansehnlichen Prototyp der ersten Generation ein formschönes und ergonomisches Gerät. Die langjährigen Erfahrungen aus den Braillezeilen, vor allem in Bezug auf die Verschmutzung und den Verschleiß der Tastoberflächen sowie die Möglichkeiten zur Reinigung und Reparatur vor Ort, flossen ebenfalls ein. So wurden z. B. die Module mit austauschbaren Tastkappen und Taststiften versehen und die Touchoberfläche aus Verschleißgründen unter die Tastkappe verlegt.

Mit diesen Modulen wurden 7 Flächendisplays aufgebaut und die Ergebnisse der Software verifiziert. Usertests wurden erfolgreich mit Soft- und Hardware durchgeführt. Befehls- und Gesteneingabe war mit den Geräten nun möglich. Die taktile Anzeige erfolgte direkt und präzise. Mit den Navigationselementen konnte einfach und effizient gearbeitet werden. Das Gewicht wurde um etwa die Hälfte auf ca. 5 kg gesenkt, die Baugröße, vor allem in der Höhe, von 100 mm auf 60 mm verringert.

Durch konstruktive Maßnahmen und den Übergang auf ein anderes Laserstrukturierungsverfahren konnten auch die Herstellkosten reduziert werden. Hier

müssen aber durch entsprechende Serienfertigung und Optimierung noch weitere Einsparungen erreicht werden.

Damit wurden die Ziele der Hardwareentwicklung im Projekt voll erreicht. In den nächsten Monaten wird dann, aus diesen Ergebnissen des Prototyps des Flächendisplays, ein Kleinserienprodukt entwickelt werden. Ab Ende 2011 sollen Displays für die Markteinführung bereitstehen.

Im Folgenden sind diese Ergebnisse der Hardwareentwicklung im Einzelnen dargestellt.

### **3.1 Geräteaufbau aus den überarbeiteten Antriebselementen aus dem Forschungsvorhaben „Interaktives Braille-Display“**

Dieses Flächendisplay besteht im Wesentlichen aus einer Tastfläche von 150 x 300 mm, aus insgesamt 720 Einzelmodulen mit je 10 Stiften und damit 7200 Taststiften, die in einer Matrix von 120 x 60 Stiften mit einem Abstand von 2,5 mm angeordnet sind. Damit ist neben der Darstellung von Graphiken, Icons, Tabellen, etc. auch die Brailleschrift darstellbar. Zu dieser Darstellung der Brailleschrift wird zwischen den einzelnen Zeichen jeweils eine Spalte von Stiften ausgelassen, so dass ein größerer Abstand zur Zeichentrennung entsteht. Nach unten wird zur Zeilentrennung eine komplette Reihe von Stiften ausgelassen. Damit kann die Tastfläche theoretisch Schrift von 12 Braillezeilen zu je 40 Zeichen gleichzeitig darstellen. Der Nutzen liegt aber in der intelligenten Nutzung von Brailleschrift mit Grafikanteilen.

Die taktile Oberfläche ist zudem mit kapazitiven Sensoren (720 Stück) zur Befehlseingabe ausgestattet. Im vorderen Bereich sind zur Eingabe und Steuerung verschiedene Taster angebracht. Die Anordnung in 2 x 5 Tastenbereiche entspricht der einer offiziellen Brailletastatur. Der modulare Aufbau bietet Vorteile im Montage- und Servicebereich.





**Abbildung 1** Stiftplatte 1 aus überarbeiteten BrailleDis-Modulen

### **3.1.1 Beschreibung des überarbeiteten Antriebselementes**

Das Einzelmodul besteht aus einer Einheit mit 10 Punkten. Diese Punkte werden durch Piezobiegewandler in Verbindung mit Keilgetrieben gesetzt und gelöscht. Im gesetzten Zustand ragen diese dann ca. 0,8 mm aus der Oberfläche heraus und sind damit tastbar. Die Biegewandler werden in die Module eingelötet und über einen speziellen Hochvoltchip angesteuert. Dieser Chip ist direkt auf dem Kunststoffkörper aufgebracht und verbunden. Die Leiterbahnen und auch der Anschlussstecker werden nach dem so genannten MID (Molded Interconnect Devices) Verfahren auch direkt auf das Kunststoffteil in der Semi-Additiv Technik aufgebracht. Die kapazitive Sensorstruktur entsteht im selben Verfahren auf dem Modulkörper. Ein spezieller Chip übernimmt die Auswertung und gibt die Signale über den Steckeranschluss an die Elektronik weiter.

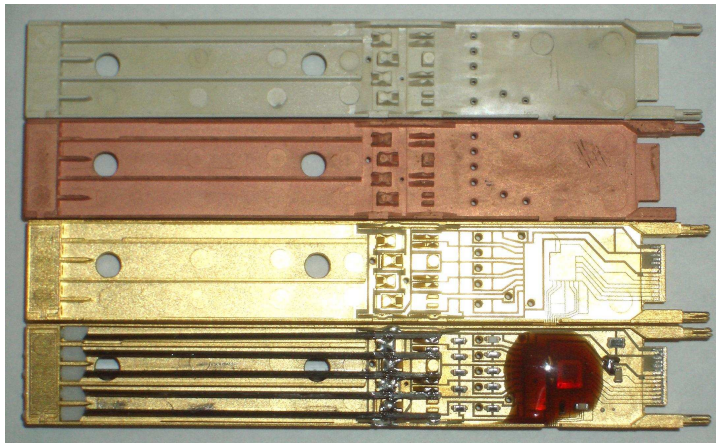


**Abbildung 2** überarbeitetes BrailleDis-Modul komplett

## a) Modulgehäuse in Semi-Additiv-Technik

Die Herstellung und Optimierung der Spritzgussteile: Die Teile haben eine äußerst komplexe geometrische Form, die an das Material, den Spritzprozess und die Werkzeugform höchste Ansprüche stellt. Das Spritzgusswerkzeug wurde mehrmals an einigen Stellen verändert und die Ergebnisse vor allem vom Institut für Zeitmesstechnik, Fein- und Mikrotechnik (IFZM) eingearbeitet. Eine genaue Auflistung der Maßnahmen ist im Bericht von IFZM enthalten. Bei den genannten Maßnahmen handelte es sich vor allem um eine Reihe von kleinen Änderungen, die aber zum Teil die Fertigung der Module erst ermöglichte bzw. zu einer höheren Prozesssicherheit bei den nachfolgenden Arbeitsschritten führte. So wurde zum Beispiel die Anbindung der 0,3 mm dicken Kontaktfahne an den Übergängen gerundet um ein Einreißen zu verhindern. Dadurch wurde die elektrische Verbindung von Backplane zum Modul sichergestellt.

Die Optimierung konnte so weit vorangetrieben werden, dass damit Module für insgesamt 8 Stiftplatten komplett aufgebaut werden konnten.



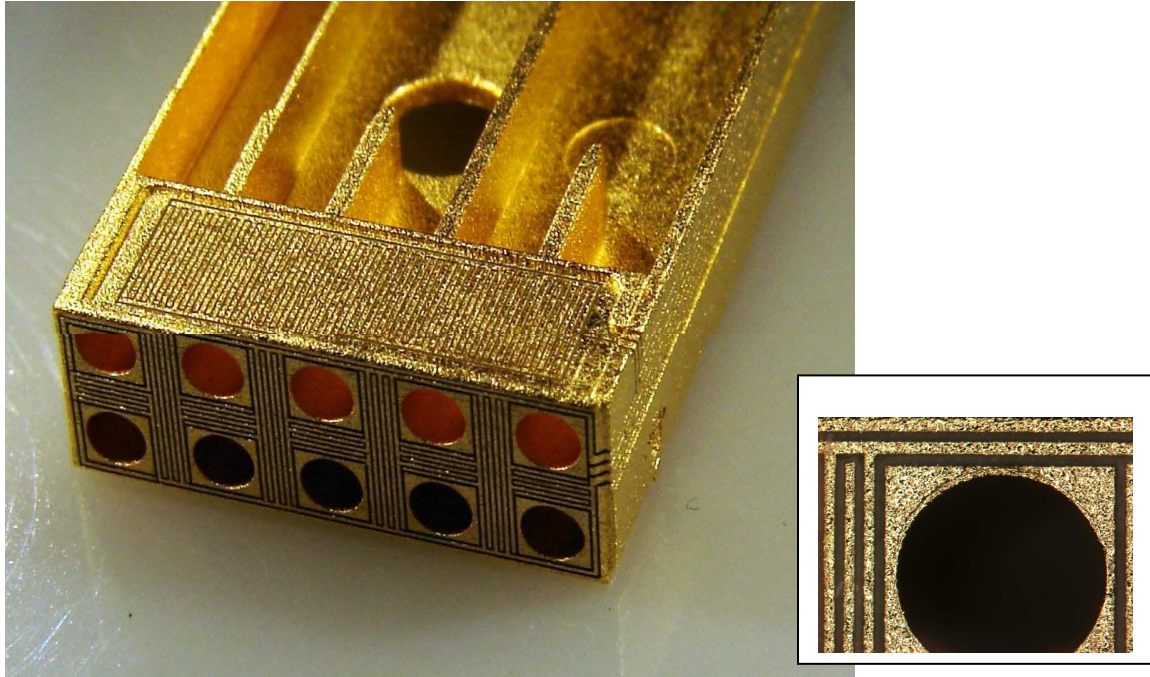
- LCP Spritzgussteil
- Spritzgussteil verkupfert
- Teil laserstrukturiert
- Modul komplett

**Abbildung 3 Modulaufbau Teilschritte**

In Abbildung 3 ist der schrittweise Aufbau der Module dargestellt. 1. Herstellung des Spritzgusskörpers, 2. Verkupferung des Gesamtteiles, 3. Laserstrukturierung der Leiterbahnen und anschließender Metallaufbau in der Galvanik, 4. Bestücken der elektronischen Bauteile, Bonden und Vergießen der Chips, Einbauen der Dots und der Biegewandler.

## b) Sensorik

Die kapazitive Struktur der Touchsensoren wird durch Laserstrukturierung im Semi-Additiv Verfahren aufgebracht. Die Leiterbahnen haben dabei nur eine Breite von 80  $\mu\text{m}$  bei einem Abstand von 70  $\mu\text{m}$ . Diese Maße stellen derzeit die absoluten Grenzen des Laserverfahrens dar. An der Seite befindet sich ein Referenzkondensator, um die Schaltung abzugleichen und Störeinflüsse zu eliminieren. Die Leiterbahnen werden dann seitlich mit dazwischen liegenden Groundleitungen nach unten zum Auswertechip geführt. Es befindet sich jeweils ein Touchsensor auf jedem Modul. Die Touchoberfläche wird nach dem Strukturieren noch lackiert, um eine gewisse Abriebsfestigkeit und Isolation zu erreichen. Hier zeigte sich jedoch, dass in der Anwendung die Module feuchtigkeitsempfindlich waren und auch die Lebensdauer in Bezug auf den mechanischen Abrieb nicht für längere Zeit ausreicht.

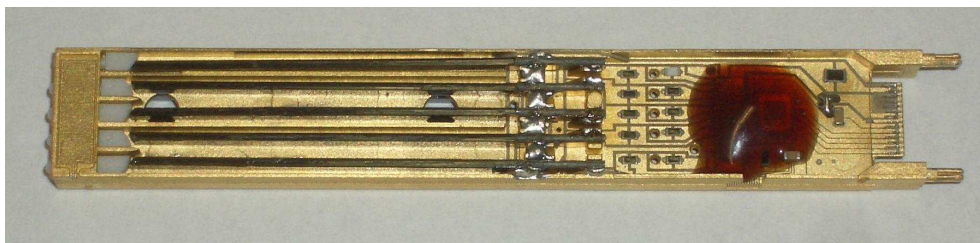


**Abbildung 4 Aufbau Sensor und Referenzstruktur**

Abbildung 4 zeigt die filigrane Sensor- und Referenzstruktur. Im Teilausschnitt ist dies nochmals größer dargestellt. Zwischen den vergoldeten Leiterbahnen darf an keiner Stelle ein Schluss erfolgen, da damit der Sensor komplett ausfällt. Dies zeigte sich als nur schwer in Serie beherrschbar.

### **c) Ansteuerung der Piezobiegewandler**

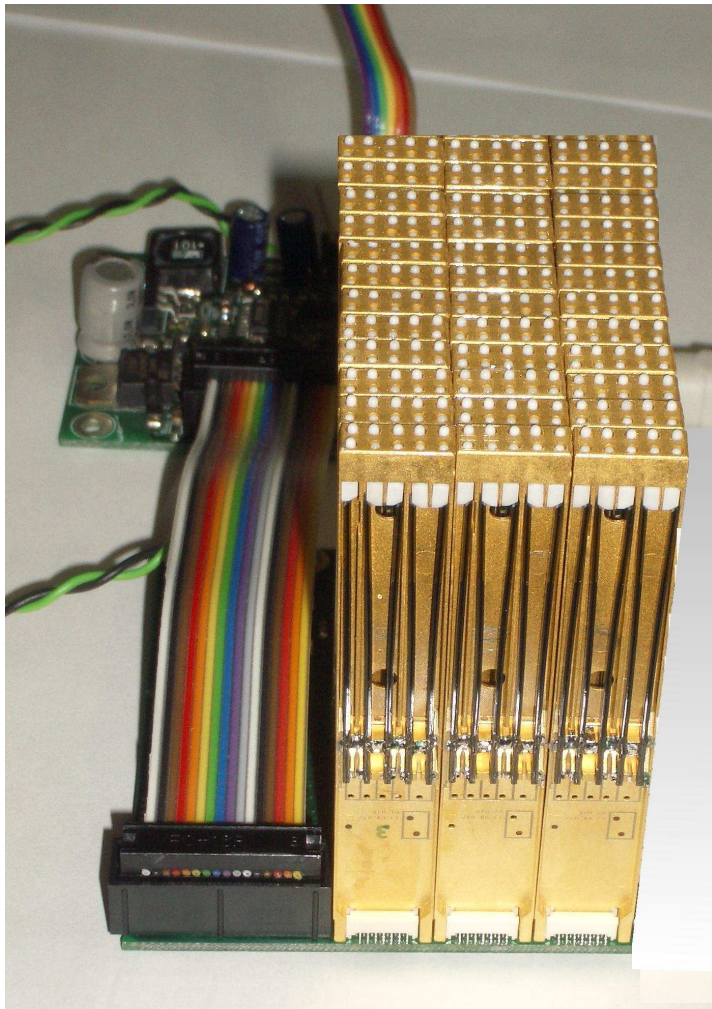
Die Piezobiegewandler werden über die Steckerfahne mit der Hochspannung von 200 V DC versorgt. Auf jedem Modul befinden sich ein Hochvolt- und ein Niedervoltchip. Der erste ist für die Ansteuerung der einzelnen Biegewandler zuständig und schaltet die 200 V auf die einzelnen Biegewandler mit einem Niedervoltsignal durch. Der Niedervoltchip beinhaltet den Logikteil und die Auswertung des Touchsensors. Damit kann über eine Busleitung jeder einzelne Biegewandler digital angesteuert werden.



**Abbildung 5 BrailleDismodul mit Chips und Steckerfahne**

In Abbildung 5 ist ein roter abgedeckter Bereich zu sehen. Darunter befinden sich die Hoch- und Niedervoltchips. Rechts daneben ist die direkt auf das Spritzgussteil strukturierte Steckerfahne zu sehen.

## d) Backplane und Datenübertragung



**Abbildung 6 BrailleDis-Testaufbau mit 48 Modulen**

Die Module haben jeweils eine serielle SPI Schnittstelle mit folgenden Signalen:

- MoSi Master out Slave in (Dateneingang)
- MiSo Master in Slave out (Datenausgang)
- CLK Clk Signal für Datenübergabe und Sensormessung
- CS Selekt Eingang, um dieses Modul auszuwählen

Die Kommunikation mit dem Modul erfolgt über ein bidirektionales 16 Bit Datenwort. Die ersten 4 Bits sind ein Befehl und bestimmen wie die nachfolgenden 12 Bit verwendet werden sollen.

Um so viele Module anzusteuern und auszulesen, ist auf der Backplane eine Dekodierungslogik implementiert. Die Module werden alle nacheinander selektiert und im Normalbetrieb werden in einem Befehl das Pinmuster geschrieben und der Sensorwert gemessen und ausgelesen. Um jedes einzelne Modul ansprechen zu können, wird die CLK Leitung für die Zeilen nacheinander freigeschaltet. Über die spaltenweise Selektierung mit dem CS Eingang des Moduls ergibt sich eine Kreuzmatrix, bei der jedes Modul einzeln angesprochen werden kann.

Zwischen dem USB- Controller und den Backplanes ist eine Busleitung, die außer der seriellen SPI Schnittstelle einen parallelen Adressbus hat, mit dem der Controller über die Dekodierungslogik der Backplane jedes Modul einzeln ansprechen kann. Die einzelnen Backplanes werden durch Lötbrücken so konfiguriert, dass sie für den Controller in unterschiedlichen Adressbereichen zugreifbar sind.

Weiterhin beinhalten die Backplanes, die jeweils 96 Module in 6 Zeilen und 16 Spalten angeordnet beinhalten, das Hochspannungsnetzteil und eine lokale 5V Stabilisierung.

Für den Betrieb der Piezobieger ist eine Spannung von 180 – 200 V notwendig, diese wird mit einem Stepupregler aus der Versorgungsspannung von 9 - 12 V generiert. Sie hat eine Strombegrenzung auf ca. 5 mA.

### **3.2 Geräteaufbau aus den neuen Antriebselementen**

Auch das HyperBraille Flächendisplay besteht aus einer Tastfläche mit ca. 150x300 mm. Die Oberfläche ist nun mit 1440 Sensoren zur Befehlseingabe ausgestattet. An den Rändern außerhalb dieser Fläche sind verschiedene Tasten zur Eingabe angebracht. Im vorderen Bereich ist eine Navigationsleiste zur schnellen und ergonomischen Bedienung zusammen mit einer Handballenaufgabe angeordnet. Cursor-Kreuze rechts und links lassen ein einfaches Verschieben und Navigieren über die Fläche zu. Insgesamt ergeben sich für diesen Prototypen Abmessungen von ca. 410 x 2,35 x 60 mm (Breite x Tiefe x Höhe). Damit ist das Gerät nur noch etwa halb so hoch wie der Vorgänger. Die aktive Touchoberfläche befindet sich nun unter so genannten Tastkappen. Damit ist sie verschleißfest und unempfindlich gegen Feuchtigkeit. Die Tastkappen sind austauschbar und reinigbar. Auch die Taststifte können bei dieser Konstruktion getauscht oder gereinigt werden, ohne das Gerät öffnen zu müssen. Damit ist dieses Gerät für den Serieneinsatz schon viel besser geeignet. Die Art und Anordnung der Tasten, Navigationsleiste und Cursorkreuze wurden unter Abstimmung der Testuser zusammen mit einem Designer für technische Geräte vorgenommen. Dadurch entstand ein formschönes ergonomisches Gehäuse. Durch seitlich angebrachte Vertiefungen kann das Gerät auch optimal transportiert werden. Mit kleinen, ausklappbaren Stützen lässt sich der Anstellwinkel der Tastfläche, ähnlich wie bei Computertastaturen, verändern. Die Ansteuerung erfolgt über einen an der hinteren Geräteseite angebrachten USB-Anschluss, die Spannungsversorgung über ein Steckernetzteil. Weitere Verbindungen sind nicht nötig.

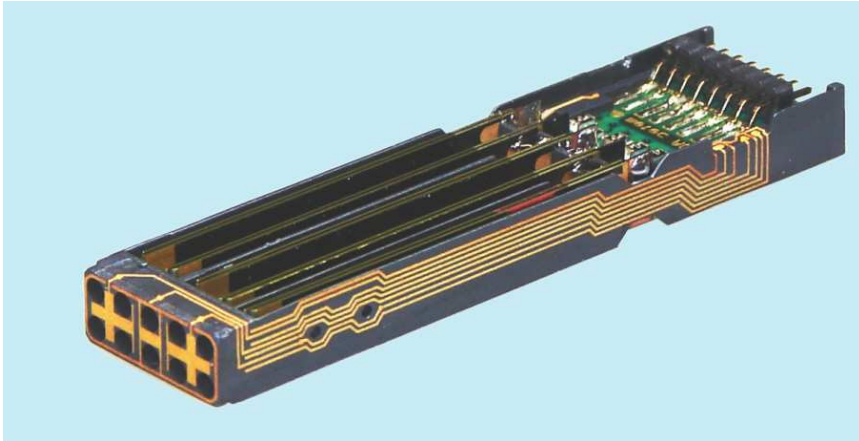


Abbildung 7 HyperBraille-Display 7200 komplett

### 3.2.1 Beschreibung des neuen Antriebselementes

Das neue Einzelmodul besteht ebenfalls aus einer Einheit mit 10 Punkten, ist aber mit einer Gesamthöhe von 60 mm um ca. 40 mm kürzer. Die Punkte werden durch Piezobiegewandler in Verbindung mit Keilgetrieben gesetzt und gelöscht. Im gesetzten Zustand ragen diese dann ebenfalls ca. 0,8 mm aus der Oberfläche heraus und sind damit tastbar. Die deutlich kürzeren Biegewandler (32,5 mm) werden in die Module eingelötet und über einen speziellen Hochvoltchip angesteuert. Diese Hochvoltchips wurden für die Module vom Institut für Mikroelektronik Stuttgart (IMS) nochmals überarbeitet. Die gesamte Elektronik ist nun auf eine herkömmliche Leiterplatte aufgebaut. Dies ergibt eine deutlich bessere Betriebssicherheit, da zum einen auf Standardprozesse zurückgegriffen werden kann (FR4 Leiterplatte mit SMT-Bestückung und Serienbonden von Chips), zum anderen kann dieses Bauteil vor der Weiterverarbeitung mit den aufwändig gelaserten Kunststoffteilen geprüft werden. Andererseits werden auch die Kunststoffteile zuvor geprüft, so dass nur Gutteile verbaut werden und damit die Ausfallquote bzw. der Verlust deutlich vermindert wird.

Die Leiterbahnen und kapazitive Struktur der Touchsensoren werden in MID (Molded Interconnect Devices) Technik nach dem patentierten Verfahren der Laserdirektstrukturierung von LPKF direkt auf das Kunststoffteil aufgebracht. Dieses Verfahren konnte nun gewählt werden, da die Struktur durch Überarbeiten des Sensorchips deutlich vereinfacht werden konnte. Dieser spezielle Chip übernimmt die Auswertung und gibt die Signale über den Steckeranschluss an die Elektronik weiter.

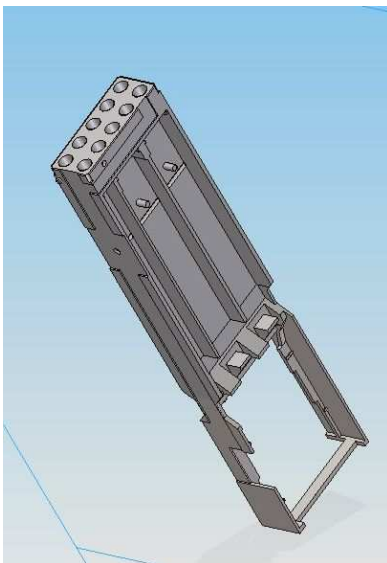


**Abbildung 8 10-Punkte Modul komplett aufgebaut**

### **a) Modulgehäuse in LDS-Technik**

Das neue Modulgehäuse wurde unter Berücksichtigung der Anwendervorgaben und der Risikobewertung komplett neu konstruiert. Entsprechend wurde eine neue Spritzgussform hergestellt. Durch den Wechsel auf das LPKF Verfahren war auch ein Wechsel auf ein anderes Kunststoffmaterial notwendig. Die Materialauswahl war hier sehr eingeschränkt, da zum einen der Kunststoff beim Spritzgießen sehr dünnflüssig werden muss, um die filigranen Formen zu füllen, und zum anderen Dotierungen für die Laseraktivierung enthalten muss. Damit konnte nur das Material LCP Vectra 840i von Ticona verwendet werden.

Die Anforderungen an das Spritzgusswerkzeug bzw. den Spritzgusskörper sind außerordentlich hoch, da bedingt durch das Laserverfahren viele Bereiche besonderen Ansprüchen genügen müssen. In Summe stellt dieses Teil höchste Anforderungen an den Spritzgusshersteller und technisch heute sicherlich das obere Ende des derzeitig Machbaren dar. Dadurch hat die seriensichere Herstellung der Spritzgussteile einige Mehrzeit und Mehraufwand benötigt als ursprünglich geplant. Das Teil kann aber nun in ausreichender Qualität hergestellt werden.



**Abbildung 9 Spritzgussteil Modulgehäuse für LDS Technik**

Der so hergestellte Spritzgusskörper wird dann, wie in

Abbildung 10 dargestellt, der weiteren Prozesskette unterzogen. Als erster Schritt erfolgt die Laseraktivierung an den Stellen, an denen die Leiterbahnen entstehen sollen. Dies ist bei diesem Verfahren die umgekehrte Technik wie beim Semi-Additiv Verfahren, bei dem mit dem Laser nicht leitende Zwischenräume erzeugt werden. Dieses Verfahren gleicht damit dem herkömmlichen Aufbau von Leiterplatten. Möglich ist dies durch Einlagerungen von Metallpartikeln im Kunststoff. Diese werden als Keime für die Metallbeschichtung durch den Laserstrahl am fertigen Spritzgussteil wieder freigelegt und damit galvanisch die entsprechenden Leiterbahnen erzeugt. Durch die Laseraktivierung von 4 Seiten des Spritzgusskörpers werden alle Leitungen dreidimensional sowie auch die Sensorstruktur erzeugt. Damit erfolgt die „Verkabelung“ der Piezobiegewandler, der Sensoren und der Leiterplatte direkt auf dem Kunststoffkörper, ohne wesentlich Raum zu beanspruchen. Nur mit solchen Techniken lässt sich bei einem Display die Rückführung der Sensorsignale auf die Anschlüsse des Moduls bewerkstelligen.

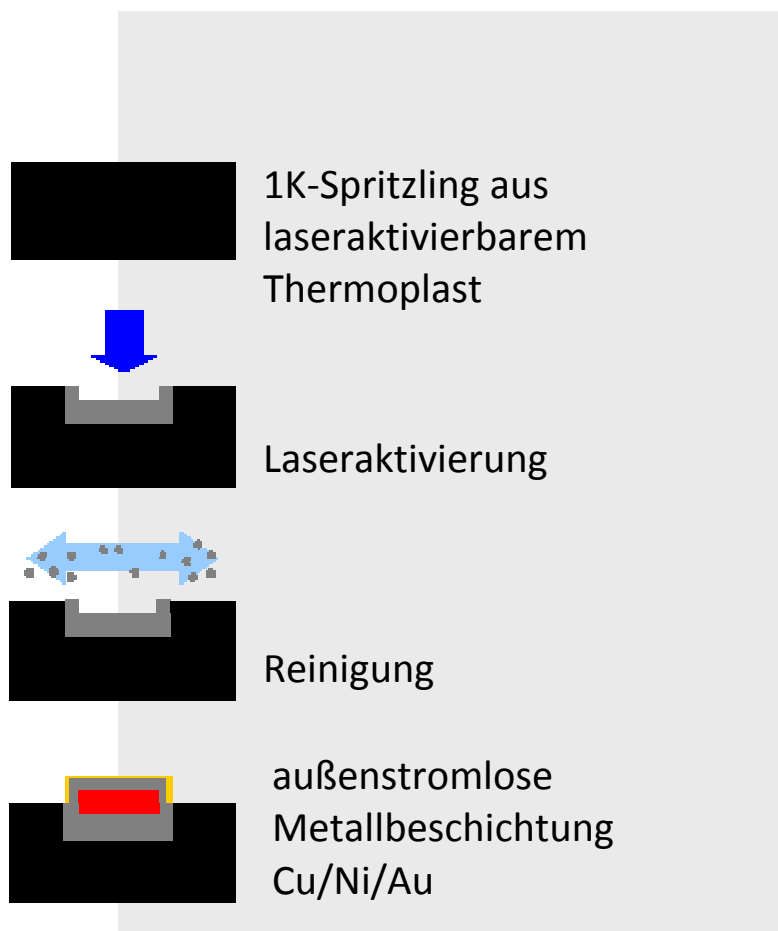
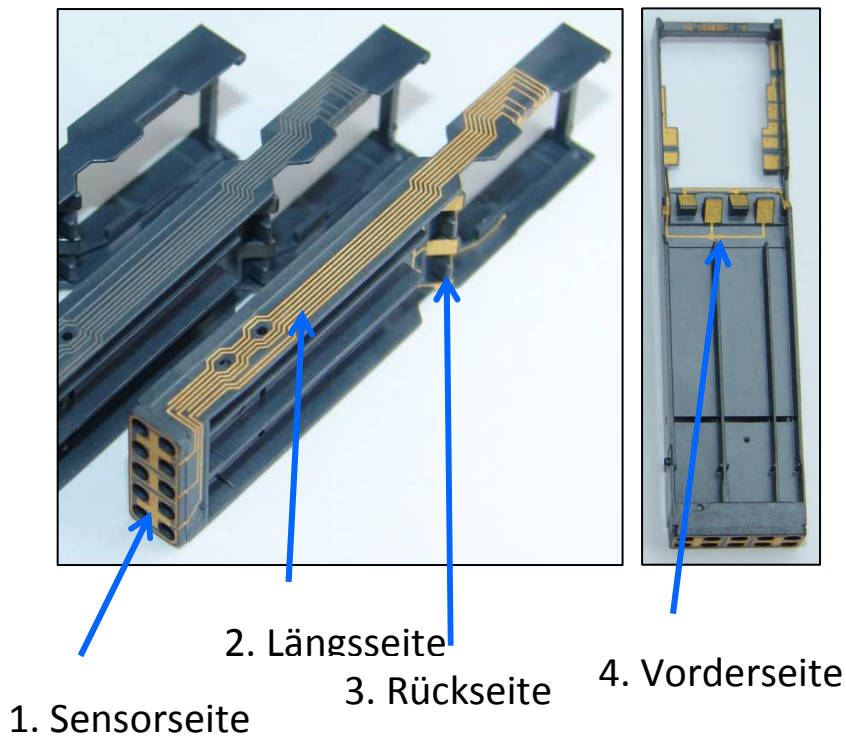


Abbildung 10 schematische Darstellung der LDS Technik



In Abbildung 11 ist der Aufbau in 3 Schritten nochmals verdeutlicht:

1. Spritzgussteile in Vectra 840 i LDS;
2. Laseraktivierung
3. galvanische Metallbeschichtung der Leiterbahnen



**Abbildung 11 Sensor- und Leiterbahnaufbau im LDS-Verfahren**

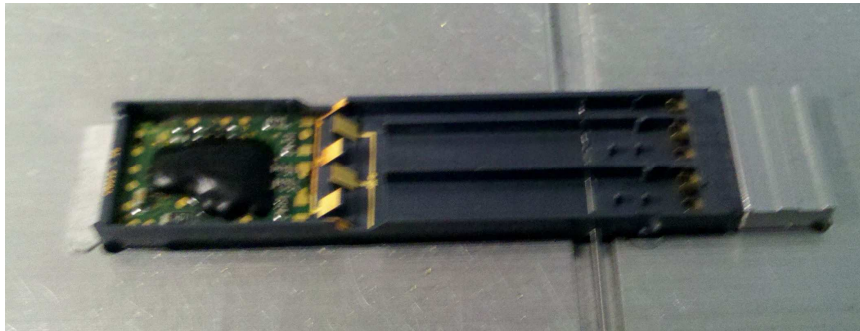
In das so vorbereitete Modulgehäuse wird nun die mit den Hoch- und Niedervoltchips bestückte und geprüfte Leiterplatte mit Leitkleber eingeklebt. In einem weiteren Arbeitsschritt wird zur besseren mechanischen Stabilität ein so genannter Underfiller zwischen Leiterplatte und Spritzgussteil eingebracht. Dieser verklebt dauerhaft die beiden Teile.

Anschließend erfolgt ein Funktionstest des gesamten Aufbaus. Durch diese Zwischenschritte wird sichergestellt, dass jeweils der vorangehende Arbeitsschritt erfolgreich war und ein weiteres Prozessieren des Teiles sinnvoll ist.

Die soweit aufgebauten Module werden anschließend, wie in Abbildung 12 dargestellt, zur Bestückung der Biegewandler in eine Montagevorrichtung gelegt. Um die Lage der Biegewandler fest zu legen, wird quer zum Modul ein metallener Distanzstift gesteckt. Dieser hält die Biegewandler bei der Montage auf konstantem Abstand zur Mittenwand. Damit streifen diese bei ihrer späteren Bewegung nicht am Kunststoffteil. Von vorne wird durch die Lagerlöcher der Pins eine Vorrichtung gesteckt, die die Biegewandler in der seitlichen Lage fixiert. Damit

wird gewährleistet, dass später im Betrieb alle Pins sicher gesetzt und gelöscht werden.

Die vorbereiteten Biegewandler werden nun in das Modul eingelegt und an den Pads angelötet. Dies geschieht derzeit in Handarbeit, da die Lötungen recht kritisch sind und die Biegewandler in der Lage noch kontrolliert werden müssen. Das so fertig gestellte Modul wird einem weiteren Funktionstest unterzogen. Hier werden die korrekten Bewegungen der 10 Piezobiegewandler und das Arbeiten der Touchsensoren überprüft. Das Modul ist nun wie in Abbildung 8 dargestellt fertig aufgebaut.



**Abbildung 12** Montagevorrichtung für die Piezobiegewandler

## **b) Sensorik**

Die Sensorik wurde gegenüber dem BrailleDis-Modul komplett überarbeitet. Zum einen wurde die Auflösung mit nun 2 Touchsensoren pro Modul verdoppelt und zum anderen die Empfindlichkeiten auf die Bedürfnisse angepasst. Vorgaben waren hier, dass auf die Module wechselbare Tastkappen aus Kunststoff montierbar sind, um im Betrieb keine Abnutzung und keinen Feuchtigkeitseinfluss zu bekommen sowie die Reinigbarkeit der Dotlöcher zu ermöglichen. Außerdem sollten die Sensoren bei gesetzten Pins ähnlich empfindlich sein wie bei gelöschten. Dies ergab eine Überarbeitung und Anpassung des Niedervoltchips in Verbindung mit einem neuen Touchsensorlayout. Dieses Layout hat den Vorteil, dass die Struktur wesentlich gröber und einfacher gestaltet werden konnte. In Abbildung 13 ist dies dargestellt. Das obere Bild zeigt die neue Sensorstruktur, das untere die des BrailleDis-Moduls. Durch bessere Abgleichmöglichkeiten im Sensorchip konnte zudem auf den aufwändigen Referenzkondensator verzichtet werden. Dies hat die Laserstrukturierung nochmals vereinfacht (Abbildung 14). Der frei gewordene Raum wurde genutzt, um die Leiterbahnen und -abstände größer zu gestalten und dem neuen LDS Verfahren optimal anzupassen.

Damit konnten die Vorgaben für die Gestenerkennung und Steuerung des Displays voll erfüllt werden. Die Eingabe über diese Sensorik erfolgt nun flüssig und betriebssicher. Die Bewegung der Biegewandler hat verhältnismäßig wenig Einfluss auf die Touchsensorik. Feuchtigkeitsempfindlichkeit oder Alterung konnten bisher nicht festgestellt werden. Für die Zukunft wäre eine etwas schnellere Abtastung wünschenswert, um vor allem schnelle Flippgesten noch auswerten zu können.

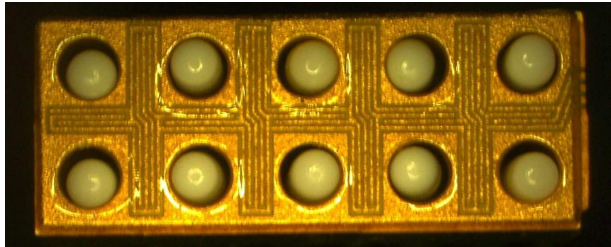
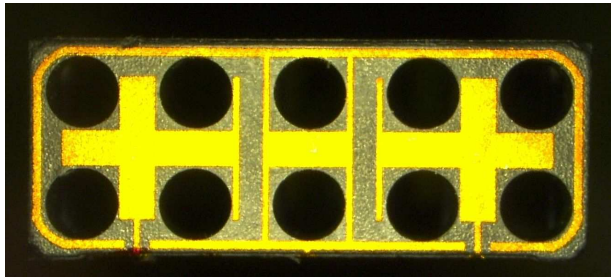


Abbildung 13 Sensorstrukturen im Vergleich BrailleDis- und HyperBraille-Modul

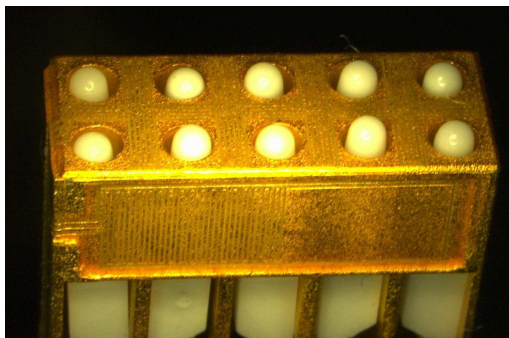


Abbildung 14 Referenzkondensator beim BrailleDis-Modul

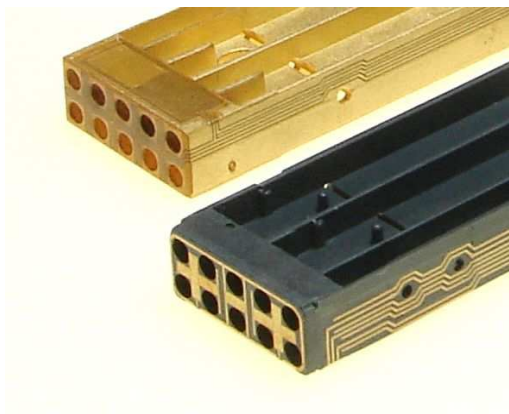


Abbildung 15 Anschlussleitungen im Vergleich

### c) Ansteuerung der Piezobiegewandler

Die Piezobiegewandler werden nun direkt über eine Pinleiste im Rastermaß 1,27 mm mit der Hochspannung von 200 V DC versorgt. Diese wird in konventioneller Bauweise auf die Platine gelötet. Damit ist der Anschluss kostengünstiger und robuster. Auf jedem Modul befinden sich wieder ein Hochvolt- und ein Niedervoltchip. Der erste ist für die Ansteuerung der einzelnen Biegewandler zuständig und schaltet die 200 V auf die einzelnen Biegewandler aus einem

Niedervoltsignal durch. Da der neue Hochvoltchip HV10d in der Projektzeit nicht funktionsfähig hergestellt werden konnte wurde ein Teil der Module mit den größeren BrailleDis-Chips (HV 10c) ausgerüstet und der verbleibende Anteil mit einer Lösung aus käuflichen 2 x 8 Kanal Chips bestückt. Durch diese Lösung war es möglich, die entsprechende Anzahl funktionierender Module aufbauen zu können.

Der überarbeitete Niedervoltchip beinhaltet den Logikteil und die Auswertungen des Touchsensors. Diese funktionierte sofort zufriedenstellend. Die Datenübertragung erfolgt über eine Busleitung, mit der jeder einzelne Biegewandler digital angesteuert wird.

## **Backplane und Datenübertragung**

Die Module haben jeweils eine serielle SPI Schnittstelle mit folgenden Signalen:

- MoSi Master out Slave in (Dateneingang)
- MiSo Master in Slave out (Datenausgang)
- CLK Clk Signal für Datenübergabe und Sensormessung
- CS Selekt Eingang, um dieses Modul anzuwählen

Die neuen Module haben einen Sensorchip, der 2 Sensoren pro Modul bereitstellt. Deshalb erfolgt die Kommunikation mit dem Modul nun über 3 bidirektional übertragene Datenbytes.

Die ersten 4 Bits sind ein Befehl und bestimmen wie die nachfolgenden 20Bit verwendet werden sollen. Um so viele Module anzusteuern und auszulesen, ist auf der Backplane eine Dekodierungslogik implementiert. Die Module werden alle nacheinander selektiert und im Normalbetrieb werden in einem Befehl das Pinmuster geschrieben und der Sensorwert gemessen und ausgelesen.

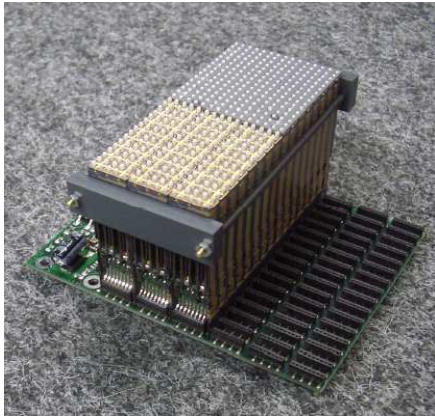
Um jedes einzelne Modul ansprechen zu können, wird die CLK Leitung für die Zeilen nacheinander freigeschaltet. Über die spaltenweise Selektierung mit dem CS Eingang des Moduls ergibt sich eine Kreuzmatrix, bei der jedes Modul einzeln angesprochen werden kann.

Zwischen dem USB- Controller und den Backplanes ist eine Busleitung, die außer der seriellen SPI Schnittstelle einen parallelen Adressbus hat, mit dem der Controller über die Dekodierungslogik der Backplane jedes Modul einzeln ansprechen kann. Die einzelnen Backplanes werden durch Lötbrücken so konfiguriert, dass sie für den Controller in unterschiedlichen Adressbereichen zugreifbar sind.

Durch die Verwendung von 3 Bytes pro Modul musste die Datenübertragung effizienter gemacht werden damit die ganze Platte in 50 ms komplett gelesen werden konnte. Dazu werden nun die obere und die untere Hälfte der Platte gleichzeitig beschrieben. Dazu sind dann 2 serielle SPI Schnittstellen nötig. Da mit der oberen und der unteren Hälfte gleichzeitig kommuniziert wird, können dieselben Adressleitungen verwendet werden.

Weiterhin beinhalten die Backplanes, die jeweils 96 Module in 6 Zeilen und 16 Spalten angeordnet beinhalten, das Hochspannungsnetzteil und eine lokale 5 V Stabilisierung.

Für den Betrieb der Piezobieger ist eine Spannung von 180 – 200 V notwendig, diese wird mit einem Stepupregler aus der Versorgungsspannung von 9 – 12 V generiert. Sie hat eine Strombegrenzung auf ca. 5 mA.



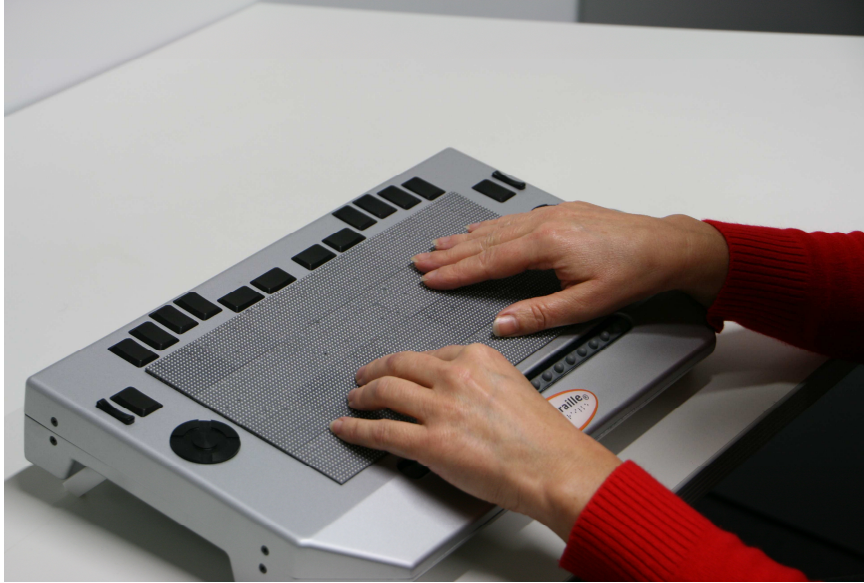
**Abbildung 16 HyperBraille-Display Testaufbau mit 48 Modulen**

### **Gehäuse**

Die Gehäuse bestehen nun aus form schönen Tiefziehteilen, die nach den Vorgaben der Usertester entwickelt wurden. Die Tastenanordnung im hinteren Bereich erlaubt auch noch eine Brailleschripteingabe. Rechts und links sind jeweils ein Tastenkreuz, eine Zoomwippe und eine HyperBrailletaste angeordnet. Diese Tasten dienen direkt zum Steuern des HyperReaders. Im vorderen Bereich wurde die Navigationsleiste der Fa. Papenmeier eingesetzt. Diese hat sich schon über lange Jahre in Braillezeilen bewährt.

### **3.3 Zusammenfassung Hardwareentwicklung**

Mit den neu entwickelten Modulen konnten Flächendisplays aufgebaut werden, die voll den Anforderungen des Pflichtenhefts gerecht wurden. Die Vorgaben der Softwareentwickler, User und des Designers wurden vollständig umgesetzt. Das Display erlangte im Projekt schon ein Stadium ähnlich einem Vorseriengerät. Alle Funktionen und Anforderungen an ein Seriengerät, wie Verschleißfestigkeit, Reinigbarkeit, Servicefreundlichkeit usw. wurden in der Konstruktion bereits berücksichtigt. Damit sind die besten Voraussetzungen für die Weiterentwicklung in ein Seriengerät erreicht worden. Die wichtigsten folgenden Schritte werden nun die seriensichere Herstellung der Module und die Reduzierung der Herstellkosten sein.



**Abbildung 17 HyperBraille-Display in Betrieb**

## **4. Ergebnisse der Softwareentwicklung**

Durch die Entwicklung des HyperReaders konnten viele neue Konzepte der Bedienung eines „Computers für Blinde“ ausprobiert werden. Die Software stellte dadurch die Grundlage für eine Vielzahl von Benutzertests dar, welche die weitere Entwicklung der Software maßgeblich beeinflusst haben.

Die Entwicklung der Software erfolgte dementsprechend in mehreren Schritten. So wurde sichergestellt, dass die Erfahrungen aus Benutzertests auf das Endprodukt Einfluss nehmen konnten und so die Bedienung optimiert werden konnte. Es zeigte sich, dass es notwendig war, die Software spätestens alle zwei Monate auf einen einheitlichen Stand zu bringen. Nur das sorgte dafür, dass die Verantwortlichen der Benutzertests ein aktuelles, einheitliches Softwarepaket hatten, mit dem neue Erfahrungen gemacht werden konnten, welche dann auch schnell in die Softwareentwicklung einfließen konnten.

Damit wurden alle Ziele der Softwareentwicklung erreicht. Die Art und Weise, wie die Software zu bedienen und zu verwenden ist, steht fest und funktioniert. Ein Blinder kann damit effizient und effektiv arbeiten. In den nächsten Monaten wird die Entwicklung insofern weitergetrieben, dass viele Bereiche innerhalb von Windows noch für den Hyperreader erschlossen werden müssen, welche bisher nicht Teil des Projekts waren, wie z. B. neue oder andere Programme, die aber trotzdem eine spezielle Unterstützung benötigen.

### **4.1 Gesamtsoftware Hyperreader**

Der HyperReader ist ein Programm, welches in Windows (Vista) im Hintergrund läuft, ohne dass der Anwender in erster Linie etwas davon mitbekommt. Die Aufgabe des HyperReaders ist es, alle Daten, die auf dem Monitor dargestellt werden, dem Blinden akustisch oder auf dem BrailleDisplay darzustellen. Das bedeutet, die Software muss in der Lage sein, die Informationen des Monitors auszulesen, diese Daten auszuwerten, die wichtigen und notwendigen Daten auszufiltern und diese Daten dem Blinden in kleinen Häppchen zu präsentieren, so dass dieser von der Informationsflut nicht überfordert wird.

Der Blinde hat dann noch die Möglichkeit, diesen HyperReader zu steuern, um die akustische und taktile Ausgabe seiner Lesegeschwindigkeit anzupassen, Dinge wiederholen zu lassen oder die Art der Darstellung zu ändern, so dass der Blinde so effizient wie möglich Daten aufnehmen kann. Zusätzlich bietet die Software die Möglichkeit, den Computer selbst zu bedienen, indem man beispielsweise mit Gesten die Maus versetzt und einen Mausklick durchführt.

#### **4.1.1 Aufbau**

Die Software besteht aus vielen einzelnen Modulen. Jedes Modul hat seine eigene Aufgabe, die es im Gesamtsystem zu erfüllen hat. Im Wesentlichen kann man diese Module in folgende verschiedene Gruppen aufteilen:

1. Filter, die die Aufgabe haben, Daten aus speziellen Programmen zu extrahieren und diese Daten den restlichen Modulen des HyperReaders zur

Verfügung zu stellen. Beispielsweise ist ein ExcelFilter dafür verantwortlich, Daten aus dem Programm Excel zu extrahieren.

2. Manager, welche die Aufgabe haben, spezielle Bereiche des HyperReaders *allumfassend* zu behandeln. Beispielsweise ist der InteractionManager für alle Interaktionen verantwortlich, die mit dem HyperReader durchgeführt werden. Der FilterManager ist für alle Filter verantwortlich. Manager sind hauptsächlich im Hauptprozess zu finden, mit dem Namen HBMonitor oder FilterServer.
3. Renderer sind für Ausgabe auf einer Hardware verantwortlich, z. B. auf dem BrailleDisplay oder als Sound.
4. AddIns. AddIns sind Programmerweiterungen des HyperReaders, die Standardverhalten für Spezialfälle abändern.
5. Sonstige – Es gibt vergleichsweise wenige Module, die nicht unter die bereits erwähnten Modulkategorien fallen.

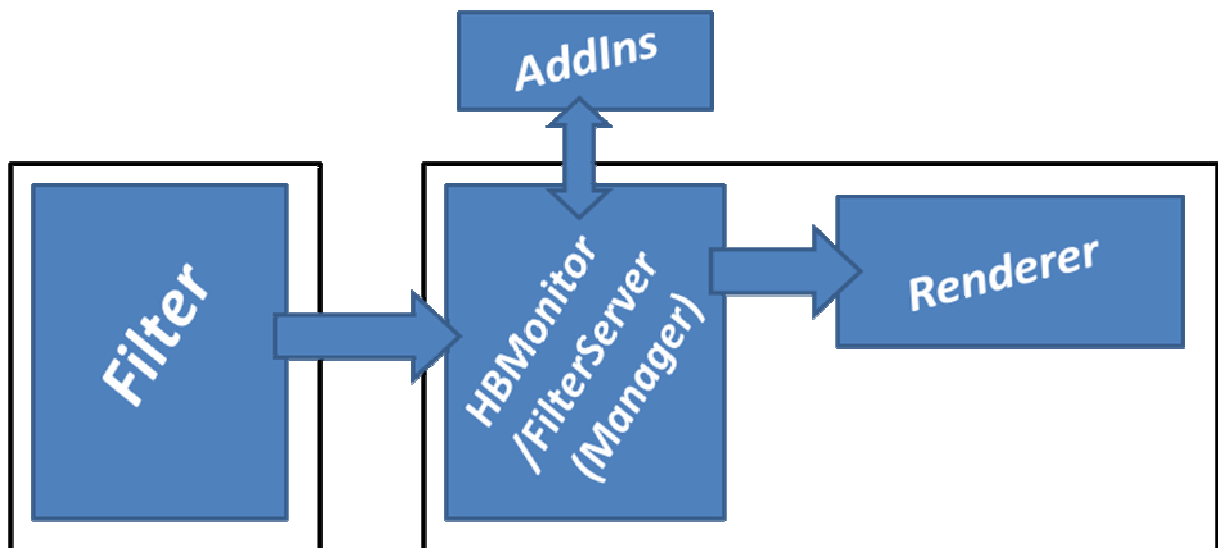


Abbildung 18 Aufbau des HyperReaders

Auf die einzelnen Softwarekomponenten wird detailliert im Kapitel 3.2 Softwarekomponenten eingegangen.

#### **4.1.2 Installation**

Zur Installation wird dem Anwender ein Setup-Programm zu Verfügung gestellt. Dies ist eine MSI-Datei, also ein von Microsoft unterstütztes Setupformat. Beim Ausführen der Setupdatei wird der gesamte Hyperreader installiert, danach ist nur noch ein Neustart notwendig.





**Abbildung 19 Screenshot einer Installation**

Das Setupprogramm wird mit WIX erzeugt. WIX steht für „Windows Installer XML“. Das Grundgerüst des Setupprogramms wurde von der Fa. Papenmeier schon Anfang 2009 erstellt. Ein Großteil der Arbeit bezüglich des Themas „Installation“ bestand danach jedoch darin, alle Änderungen aller Partner regelmäßig in das Setupprogramm einzupflegen. Das Setup war ein wesentlicher Teil des Arbeitspakets 16.

### **Abschlussbericht AP16 - Projektzusammenfassung**

Ziel des AP16 war es, für eine zusammenhängende Software zu sorgen. Ohne dieses Paket gäbe es sonst einen Satz verschiedener Module, die nicht zusammenarbeiten könnten und somit gar nicht brauchbar wären.

Folgende Ziele wurden für das AP 16 bestimmt

1. Zum Projektende muss ein Setup erstellt sein, das ein lauffähiges HyperReader-Programm installiert. Dieses Setup soll alle Module enthalten, die im Rahmen des HyperBraille-Projekts entwickelt wurden.

Im Laufe der Projektlaufzeit entwickelten sich weitere Punkte

1. Es musste während der Projektlaufzeit regelmäßig neue aktuell gehaltene Setups entwickelt und angepasst werden, um die verschiedenen Bereiche des Projekts weiterhin am Laufen zu halten.

## **Konzept des Setups**

Das Setup ist im Wesentlichen eine Ansammlung aller Programmbibliotheken und Programme, die im Projekt von allen Partnern erzeugt wurden. Die Ziele des Arbeitspakets sind dabei immer:

- 1) Bei der Installation die Dateien an die richtigen Stellen kopieren
- 2) Registriervorgänge durchführen.
- 3) Unstimmigkeiten zwischen einzelnen Modulen beheben.

## **Ablauf der Entwicklung des Projektzusammenfassung**

Da einzelne Module erst entwickelt werden mussten, bevor diese in einer Projektzusammenfassung zusammengefasst werden konnten, startete die Entwicklung dieses Arbeitspaket erst Ende des ersten Halbjahres 2009. Das Setup an sich war schnell erstellt, jedoch war sofort klar, dass eine regelmäßige Aktualisierung des Setups dringend notwendig war. Die meiste Arbeitszeit wurde in diesem Projekt sozusagen damit verwendet, das Setup auf dem aktuellen Stand zu halten.

So wurde im zweiten Halbjahr 2009 damit begonnen, regelmäßig alle 2 Monate 2 Setups zu erstellen. Diese Setups wurden dann für Funktionstest und Benutzertest verwendet und boten immer komplette, abgeschlossene Programme. Dies ergab den Vorteil, dass sowohl Funktionstest als auch Benutzertest auf eindeutigen Ständen durchgeführt wurden. Dies erleichterte sowohl die Kommunikation als auch die Zusammenarbeit mit den Mitarbeitern, welche keine Software entwickelten, sondern diese nur auswerteten. Es gab zwischen Juli 2009 und April 2011 insgesamt **25 (intern) offizielle Setups**, die in Benutzertests verwendet wurden. Zusätzlich wurden etwa nochmal **25 individuelle Setups** entwickelt, mit denen manche Entwickler ihre persönlichen Arbeiten am Gesamtsystem testen konnten.

### **4.1.3 Handhabung**

Nach der Installation ist das Programm selbststartend. Das heißt, durch das Booten des Windows-PCs sorgt man schon dafür, dass das Programm gestartet wird. Dies bringt den wichtigen Vorteil mit sich, dass ein blinder Anwender sofort den Computer bedienen kann, ohne dass er gezwungen ist, mit gut Glück das richtige Programm zu starten. Desweiteren war es nur so möglich, Zugriff auf den Login-Bildschirm von Windows zu erhalten („Drücken Strg+Alt+Entf zum Anmelden“).

Es wurden zwei Versionen des Hauptprogramms entwickelt: Der sogenannte Filterserver, welcher automatisch bei Computerstart starten kann, und der sogenannte HBMonitor. Der HBMonitor ist das Programm, welches ebenfalls den HyperReader startet, jedoch bietet es für den Entwickler noch ein paar Ansichten der laufenden Prozesse. Mit dieser Ansicht wurde es ermöglicht, dass ein Entwickler schneller und besser Fehler finden konnte. Außerdem war mit dem HBMonitor der Entwickler in der Lage, auch ohne ein echtes BrailleDisplay zumindest zeitweise zu arbeiten.

Der HBMonitor wurde soweit verbessert, dass er auch zu verschiedenen Anlässen bei Live-Präsentationen Verwendung gefunden hat.

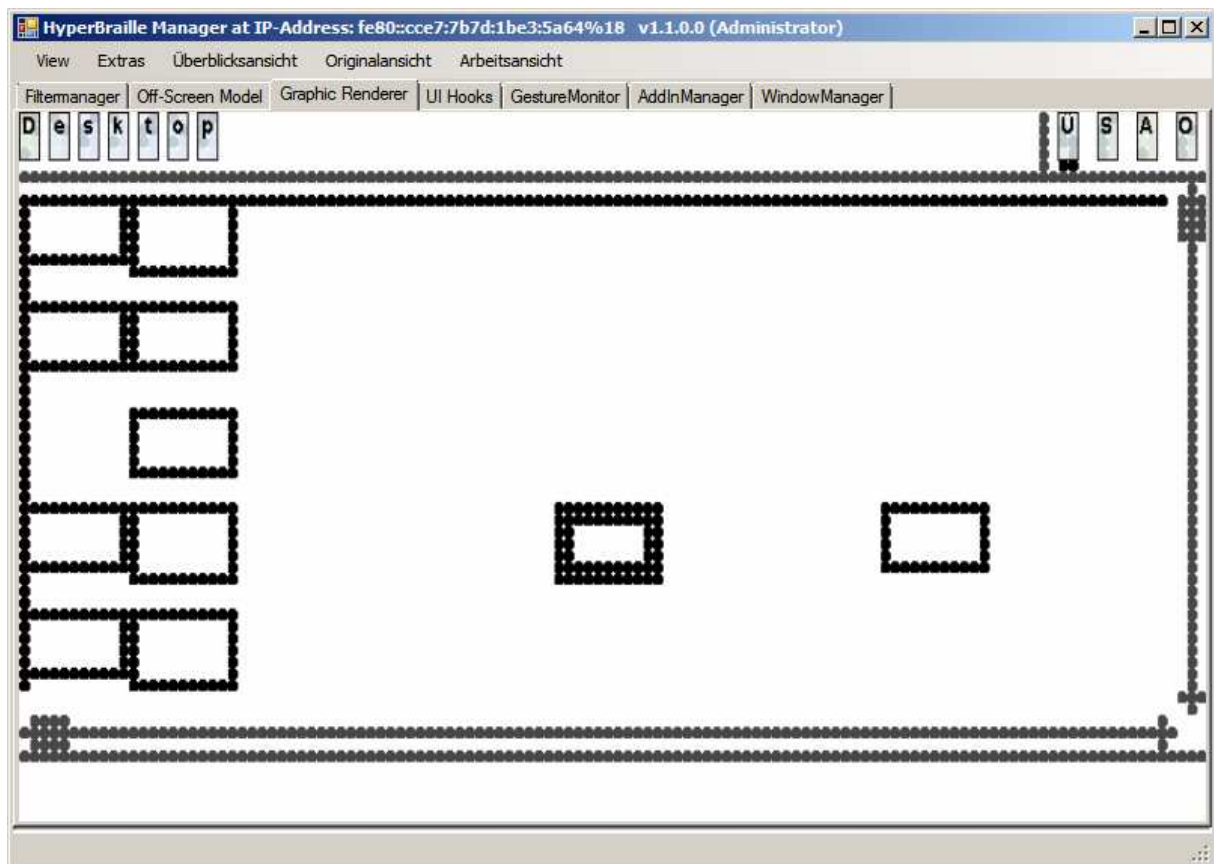


Abbildung 20 Ansicht des HBMonitors: Das Bild zeigt den Zustand des Braille-Displays, um den Desktop in der Überblicksansicht darzustellen.

Im Abschlussbericht des Arbeitspakets 2 „Entwurf Bedieneroberfläche und Anwenderfunktionalität“ der Technischen Universität Dresden und der Blindenstudienanstalt e. V. Marburg wird bezüglich der Handhabung detaillierter eingegangen. Im Wesentlichen wurden alle notwendigen Vorgaben aus eben diesem Arbeitspaket in die Software integriert.

## **4.2 Softwarekomponenten**

Die HyperReader-Software besteht aus vielen einzelnen Komponenten. Abbildung 21 zeigt den Überblick über fast alle Komponenten und deren Zuordnung zu den verschiedenen Arbeitspaketen.

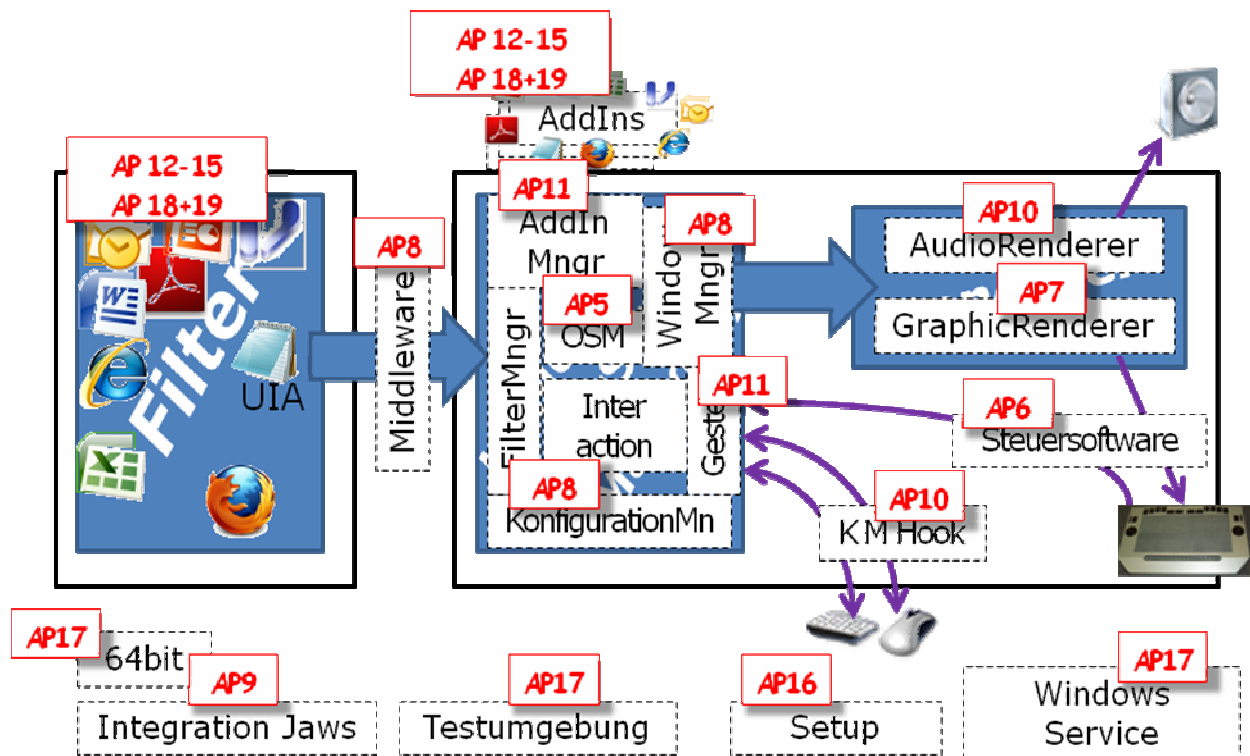


Abbildung 21 Überblick über Softwaremodule und deren Zuordnung zu verschiedenen Arbeitspaketen

Hier ein kleiner Überblick über die einzelnen Komponenten:

- Filter. Die Filter sind verantwortlich, Daten aus einzelnen Programmen zu extrahieren. Für viele einzelnen Programme musste je ein Filter geschrieben werden: Word (UniP), PowerPoint (TUD), Visio (TUD), Excel, InternetExplorer (UniP), Mozilla Firefox (TUD/Vis), Outlook, Adobe Acrobat (TUD). Nur die Filter mit den Namen „UJA-Filter“ (TUD) und „MSAA-Filter“ sind unabhängige Filter, die grundsätzlich mit allen Programmen zusammenspielen, dann aber nur eingeschränkt funktionieren.
- Addins. Für jedes Programm wurde, wie auch bei den Filtern, eine Spezialanpassung geschrieben, die sich in den AddIns widerspiegelt.
- Middleware. (TUD) Die Middleware ist eine Schnittstelle zur Kommunikation der Filter mit dem Rest des HyperReaders. Dies war notwendig, da die Filter im Normalfall in einem anderen Prozess laufen als der restliche HyperReader.
- FilterManager. (TUD) Dies ist eine Komponente, welche notwendige Filter startet und diese überwacht.
- InteractionManager. (TUD) Dieser überwacht und kontrolliert alle Eingaben des Anwenders: Gesten, Tastendrücke, etc.
- GestenManager (TUD): Dieser wertet die Berührungen des Anwenders aus. Wird eine Geste erkannt, so wird diese an den InteractionManager weitergegeben.

- KonfigurationsManager (TUD): Dieser stellt Mechanismen zur Verfügung, die eine flexible Konfiguration für den HyperReader ermöglichen.
- WindowManager (TUD) . Dieser überwacht wo welches Fenster dargestellt wird und welche Fenster in Windows offen sind und den Fokus haben.
- OSM-Manager (TUD). Dieser stellt eine HyperReader-interne „Datenbank“ zur Verfügung, in der alle Daten gespeichert werden, die von den Filtern aus den Windowsprogrammen extrahiert wurden.
- AddInManager (TUD): Dieser überwacht alle AddIns
- AudioRenderer: Diese Komponente ist dafür verantwortlich, Informationen per Sprache über den Audioausgang auszugeben.
- GraphicRenderer (TUD/Vis): Diese Komponente ist dafür verantwortlich, die darzustellenden Daten in eine „monochrome Stiftmatrix“ zu wandeln, damit diese dargestellt werden kann.
- Steuersoftware. Die Steuersoftware umfasst alle geräteabhängigen programmierten Komponenten zur Ansteuerung des BrailleDisplays. Dazu zählen z. B. Treiber und Firmware.
- KMHook, bzw. KeyboardMouseHook: Diese Komponente ist dafür verantwortlich, die Tastatur und die Maus des PCs anzusprechen und diese ggf. direkt zu steuern.
- 64Bit: Dies ist keine Komponente der Software, aber eine eigene Schwierigkeit, die speziell betrachtet werden musste. Die Software hat unterschiedliche Anforderungen, je nachdem, ob der Anwender ein 64bit oder ein 32bit Betriebssystem hat.
- Integration Jaws (Papenmeier): Jaws ist ein Screenreader von Freedom Scientific. Dieser Screenreader hat sich in der bisherigen Blindenwelt durchgesetzt, jedoch ist er nicht in der Lage, eine Stiftplatte mit zweidimensionalen Daten anzusteuern. Eine Integration dieses Screenreaders sollte blinden Anwendern einen leichteren Übergang ermöglichen.
- Setup (Papenmeier/Metec): Das Thema Setup wurde schon im Kapitel Installation behandelt.
- Windows Service: Damit der Hyperreader jederzeit im Hintergrund arbeiten kann, musste dieser als Windows-Hintergrund-Service aufgebaut werden.

Im Weiteren wird auf die Softwarekomponenten eingegangen, für die Metec selbst verantwortlich war.

#### **4.2.1 Softwarekomponente Stiftplattenansteuerung – AP6**

Ein Ziel von Arbeitspaket 6 ist es, dem HyperReader einen Zugriff auf die Stiftplatte zu ermöglichen.

Folgende Unterziele wurden in der Arbeitspaketbeschreibung definiert:

1. Entwicklung einer Firmware innerhalb des Gerätes
2. Entwicklung eines Windows-Treiber
3. Entwicklung einer .Net-fähigen Treiberansteuerung

Im Projektverlauf wurden zusätzlich folgende Punkte implementiert:

1. 64-Bit Kompatibilität
2. Anpassungen an die zweite Stiftplattenversion
3. Mehrfache Anpassung der Firmware für die verschiedenen verarbeiteten Chips auf den Modulen, so dass das Gerät auch verschiedene Chips im Parallelbetrieb laufen lassen kann.

### Konzept der Stiftplattenansteuerung

Innerhalb des HyperReaders wurde das AP6, die Stiftplattenansteuerung, hauptsächlich von zwei anderen Komponenten genutzt. Der GraphicRenderer (AP7) benötigt die Stiftplattenansteuerung, um die gewünschte Darstellung auf der Stiftplatte anzuzeigen. Der InteractionManager (AP11) benötigt die Stiftplattenansteuerung, um alle Eingaben, die der Anwender durch die Stiftplatte tätigt, zu empfangen, beispielsweise Tastendrucke oder Gesteneingaben.

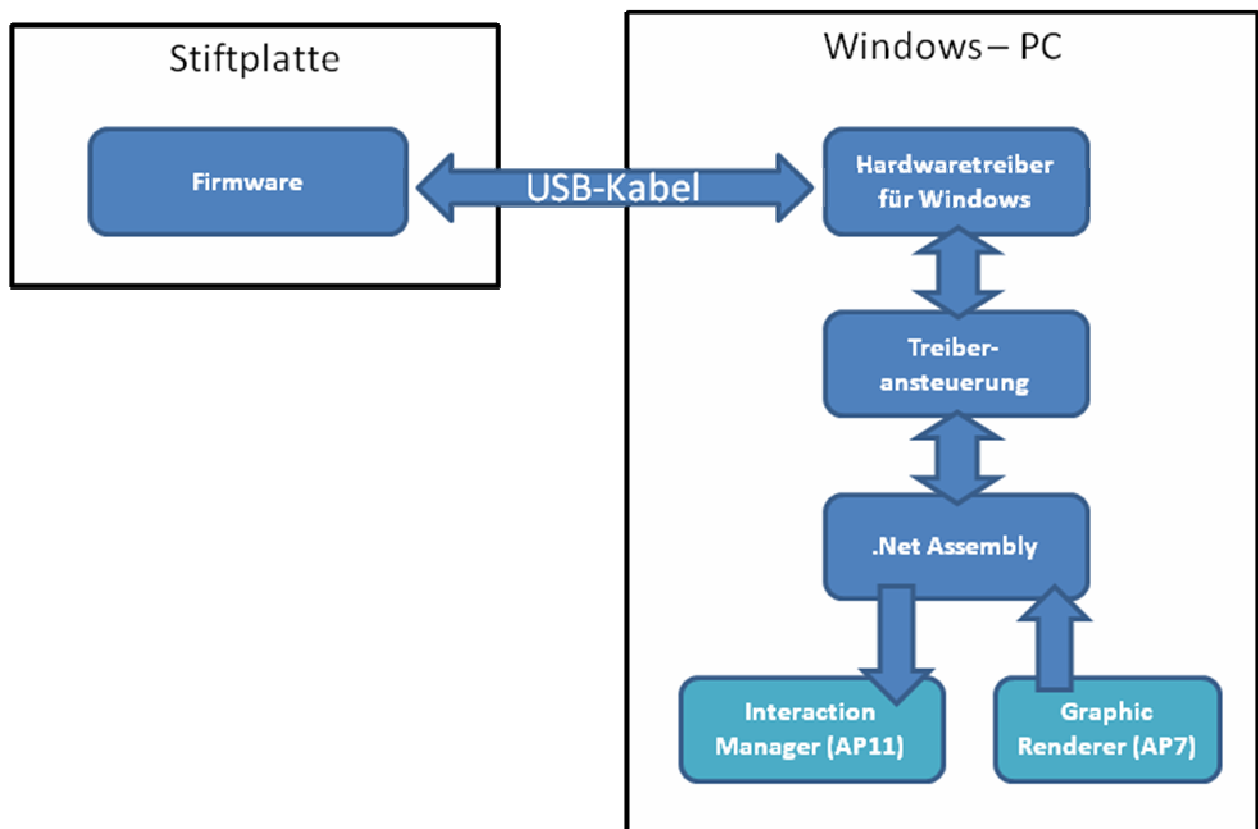


Abbildung 22 Aufbau der Stiftplattenansteuerung

Abbildung 22 veranschaulicht das Zusammenspiel der Komponenten. Die Firmware ist eine eigenständige Software, welche direkt im Gerät die Hardware steuert und die Sensoren abfragt. Über das USB-Kabel kommuniziert das Gerät/Firmware mit Windows. Um die Informationen des Geräts verwenden zu können, musste für Windows ein Treiber entwickelt werden, der wie jeder andere Gerätetreiber installiert werden muss. Die „Treiberansteuerung“ und das „Net Assembly“ sind beides Programmibliotheken, welche die Treiberfunktionen den anderen Modulen zur Verfügung stellen. Eine Unterscheidung zwischen beiden wurde deswegen gemacht, weil sich die „Programmierwelten“ deutlich unterscheiden. So ist die „Treiberansteuerung“ in C++ programmiert um der Schwierigkeit der Treiberansteuerung gerecht zu werden. Das „Net Assembly“ ist jedoch in C# programmiert, um sich dem Rest des HyperBraille-Projektes anzupassen und die Verwendung der Stiftplattenansteuerung zu vereinfachen.

### **Ablauf der Entwicklung der Stiftplattenansteuerung**

Nach Start des AP 10 (Halbjahresbericht 07/2008) wurde gleich die unterste Kommunikationsschicht realisiert. D.h. die Firmware war dann bereits so weit, nach außen zu kommunizieren und intern die Hardware anzusteuern. So war die Firmware in der Lage, sowohl Tasten, Touchwerte und Stifte anzusteuern. Treiber und Treiberansteuerung waren bereits so weit, dass Stifte auf der Stiftplatte gesetzt werden konnten.

Im zweiten Halbjahr 2008 wurde das .Net-Assembly erstellt, um die Mechanismen im HyperReader verwenden zu können. Die Firmware wurde für neue Chips angepasst. Zusätzlich wurde die Abtastrate für die Braille-Tasten auf 100 Hz erhöht, um diese auch wirklich als Brailletastatur verwenden zu können. Die bisherigen 20 Hz waren zu langsam. Der Treiber, die Treiberansteuerung und die .Net Assembly waren in der Lage, alle Mechanismen umzusetzen. Zusätzlich wurden große Fortschritte und Verbesserungen in Zugriffsgeschwindigkeit und Zuverlässigkeit erreicht.

Im ersten Halbjahr 2009 wurde auch der 64Bit-Treiber für die Stiftplatte entwickelt. Entwickler mit einem 64-Bit Betriebssystem konnten die Stiftplatte nun auch verwenden. Ansonsten gab es Versuche und Tests für die Empfindlichkeitssteuerung der Touchwerte der Stiftplatte.

Im zweiten Halbjahr 2009 ruhte die Arbeit für die Stiftplattensteuerung, da das Arbeitspaket abgeschlossen war. Durch die Verlängerung des Projekts und die Neuentwicklung eines neuen Stiftplattentyps wurde Anfang 2010 damit begonnen, die Anpassungen für den neuen Stiftplattentyp vorzunehmen. Dazu gehörte es, ein neues Protokoll zu entwickeln und die Änderungen trotzdem so flexibel zu gestalten, dass auch die bisherigen Stiftplatten noch weiterhin verwendbar waren. Im zweiten Halbjahr 2010 blieben nur noch Fehlerverbesserungen, da ansonsten das Arbeitspaket abgeschlossen war.

#### **4.2.2 AudioRenderer, Tastaturhook, etc. - AP10**

Das Arbeitspaket 10 gliedert sich in einige verschiedene Bereiche:

AudioRenderer, TastaturMausHook, MSAA-Filter, Videofilter.

Nicht alle dieser Unterpunkte wurden auch vollständig behandelt.

## **Videofilter**

Das Ziel des Videofilters war es, mit einem eigenständigen Graphiktreiber die Zeichenbefehle von Windows abzufangen, um diese dann in Informationen zu wandeln, die der Hyperreader darstellen kann. Jedoch erkannte man schon Mitte 2008, dass die Verwendung des Videofilters durchaus fraglich ist. Anfang 2009 war es sicher, dass eine Entwicklung des Videofilters überflüssig ist. Im Wesentlichen bestand die Argumentation aus zwei Fakten:

- 1) Die Windowsfunktionalität zum Erzeugen von Screenshots war in jeder Hinsicht sehr gut. Eine Möglichkeit zum Abfangen von Zeichenereignissen am Bildschirm war nicht mehr notwendig.
- 2) Das Verhalten von Vista (und auch Windows7) war in Hinsicht der Erwartungen für einen Videofilter nicht mehr brauchbar. Die Idee des Videofilters mit Hilfe eines Graphiktreibers stammte aus der Zeit von Windows XP. Unter XP wäre man dadurch leicht an mehr Informationen als nur die eines Screenshots gekommen. Das Verhalten von Windows hatte sich bei Vista jedoch geändert, wodurch die Idee nicht mehr brauchbar war.

## **MSAA-Filter**

Ziel des MSAA-Filters war es, aus beliebigen Anwendungen und aus Windows Vista selbst Informationen zu extrahieren. MSAA steht für MicroSoft Accessibility API, also eine Technik, die einzelne Fensterelemente zugänglich machen soll. Neben MSAA bietet Microsoft inzwischen eine neue bessere Technik an: UIA.

Die Informationen und Ereignisse, welche MSAA liefert, soll in Objekte des OSM (AP 5) gewandelt und im OSM-Manager gespeichert werden.

### **Ablauf der Entwicklung des MSAA-Filters**

Im ersten Halbjahr 2008 wurde der MSAA-Filter in seinen Grundlagen erstellt. Er konnte die MSAA-Elemente aus Windows auslesen, und auf Ereignisse reagieren.

Im zweiten Halbjahr 2008 war der wesentliche Punkt, dass die Informationen und Ereignisse in das HyperBraille interne System gewandelt wurden: Das OSM.

Im ersten Halbjahr 2009 wurde der Filter insofern erweitert, als dass einige wichtige Systeminformationen ausgelesen wurden, dazu gehörten z. B. Prozessstarts und Informationen aus dem Desktop oder der Taskbar. Allerdings zeigten bereits Tests mit der Konkurrenztechnik UIA, dass die veraltete Technik MSAA nicht so oft ans Ziel führt. MSAA wird von Microsoft nicht mehr unterstützt, und dies machte sich bemerkbar. Der UIA-Filter erzielte deutlich bessere Ergebnisse. Dadurch, dass die Ergebnisse deutlich besser waren, bauten die meisten programmspezifischen Filter dann auch auf dem UIA-Filter auf (z. B. Word, Excel, etc.). Der MSAA-Filter wurde obsolet.



Stand heute: Der MSAA-Filter existiert noch. Er extrahiert auch noch die Daten aus Windows. Da für den MSAA-Filter jedoch keine praktische Verwendung gefunden wurde, wurde seine Weiterentwicklung aufgegeben. Dies führte dazu, dass der MSAA-Filter seit Mitte 2009 keine Pflege mehr erfahren hat, weswegen er an einigen Stellen nicht mehr kompatibel zum aktuellen Hyperreader ist.

Die Erkenntnis durch die Entwicklung des MSAA-Filters lautet: Es ist durchaus sinnvoll, MSAA für einzelne spezielle Implementierungen zu verwenden. Ein allgemeiner MSAA-Filter, der für alle allgemeinen Programme funktionieren soll, ist sinnlos, UIA ist die bessere Technik.

## **Entwicklung einer dynamischen Audioausgabe zur Unterstützung des HyperReaders**

Ein Ziel von Arbeitspaket 10 ist es, eine Audioausgabe zur Unterstützung des HyperReaders zu entwickeln. Unter anderem wird damit dem Off-Screen-Modell (AP 8) eine akustische Grundfunktion zur Verfügung gestellt. Des Weiteren werden der Interaktion Manager (AP11) und der WindowManager (AP8) auch darin unterstützt. Diese akustische Ausgabe hilft dem Blinden bei der Bedienung dieser Elemente.

Folgende Unterziele wurden in der Arbeitspaketbeschreibung definiert:

4. Ermöglichung einer Mehrsprachigkeit.
5. Verwendung des Sprachausgabestandards SSML
6. Konfigurierbarkeit. Blinde haben bei der Sprachausgabe einen hohen Anpassungsbedarf. Dazu zählen Geschwindigkeit der Ansage, Lautstärke und Tonlage, verschiedene Stimmen. Einzelne Fenster-Elemente müssen speziell anpassbar sein.
7. Steuerbarkeit. Die Sprachausgabe muss vom Blinden kontrollierbar sein, was eine intensive Zusammenarbeit mit dem InteraktionManager (AP11) bedeutet.
8. Ein Tastaturecho, das eingegebene Zeichen direkt ausgibt.

Im Projektverlauf wurden zusätzlich folgende Punkte implementiert:

4. Erweiterbarkeit durch AddIns.
5. Gleichzeitige Ansteuerung mehrere Stiftplatten
6. Speicherung der Ausgabe zu Logging-Zwecken
7. Der Schnelllesemodus, der berührte Elemente auf der Stiftplatte direkt akustisch ausgibt.
8. Eine Profilverwaltung, die dafür sorgt, dass die Sprachausgabe von der Situation abhängig konfiguriert werden kann.

## Konzept des AudioRenderers

Der AudioRenderer ist für die Audioausgabe des HyperReaders zuständig. Jedes akustische Signal wird durch den AudioRenderer erzeugt. Das fängt an mit dem Fehlersound bei Fehleingabe und endet bei kompliziert zusammen gestellten Sätzen für Hilfestellungen bestimmter Windowselemente. Der AudioRenderer erzeugt dazu jedes Mal ein sogenanntes SSML-Dokument, welches den auszusprechenden Text beschreibt. SSML ist ein Standard wie HTML im Internet. Während HTML jedoch die Darstellung einer Internetseite beschreibt, beschreibt SSML die Art und Weise für die Sprachausgabe, also Stimme, Tonlage, Geschwindigkeit usw.

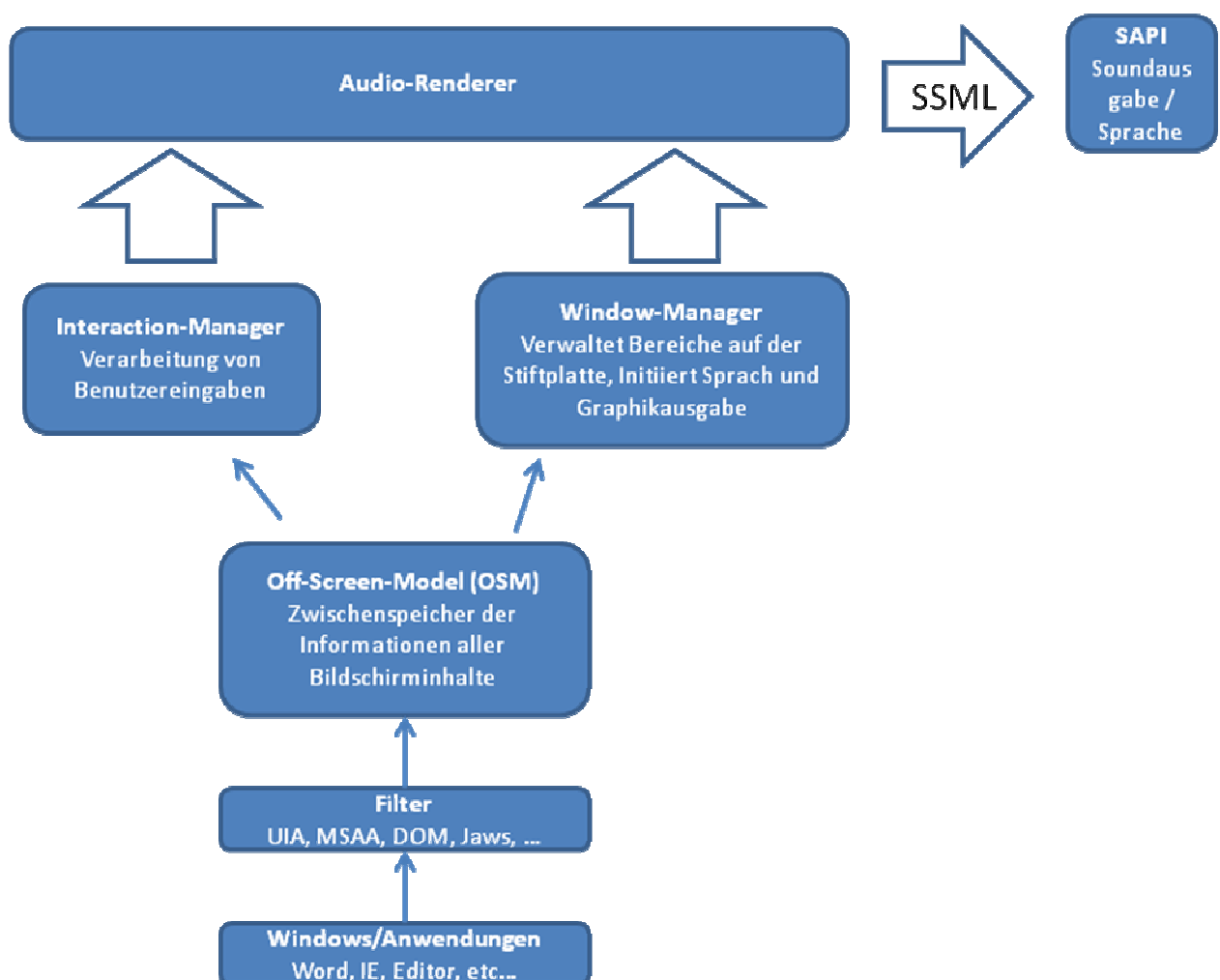


Abbildung 23 Aufbereitungs pipeline für die Audioausgabe

Der AudioRenderer ist, wie auch z. B. der GraphicRenderer, das letzte Glied in der Kette der Aufbereitungs pipeline. Audioausgaben müssen jedoch immer vom Anwender initiiert werden, um ein dauerhaftes Sprechen der Audioausgabe zu vermeiden. Grundsätzlich gibt es dazu zwei Wege:

1. Eine Windowsanwendung wird (durch die Bedienung des Anwenders) verändert. Veränderungen von Windows-Anwendungen werden über eine Vielzahl von in der HB-Software integrierten Filtern aufbereitet und in einem OffScreen-Modell (OSM) abgelegt. Das OSM verwaltet die gefilterten Inhalte in einer Baumstruktur und informiert andere Komponenten über Änderungen des Bildschirminhalts. Der OSM-Manager informiert den InteraktionManager und den WindowManager über die Änderung. Der InteraktionManager oder der WindowManager initiiert dann den AudioRenderer und beauftragt die notwendige akustische Ausgabe. Beispiel: Der Anwender drückt die Tab-Taste, um auf den nächsten Schalter in der Anwendung zu springen. Dadurch gibt es einen Fokuswechsel. Am Ende gibt der AudioRender den Inhalt des neuen Schalters wieder).
2. Der Anwender kann auch direkt Eingaben für den InteraktionManager tätigen. Solche Eingaben müssen ein akustisches Signal zurückliefern, um dem Blinden die Richtigkeit seiner getätigten Eingabe zu bestätigen. Beispiel: Beim „nach rechts scrollen“ kommt der Fehlersound – der Blinde weiß sofort, dass er bereits ganz nach rechts gescrollt hat.

Die akustische Ausgabe ist in drei verschiedene Arten aufgeteilt:

1. Die Wiedergabe von OSM-Strukturen. Von Filtern gefilterte Bildelemente müssen akustisch wiedergegeben werden.
2. Die HyperReader Sprachausgabe, wie z. B. Statusmeldungen des Screenreaders (Zoomstufe) oder auch Sounds.
3. Das Tastaturecho

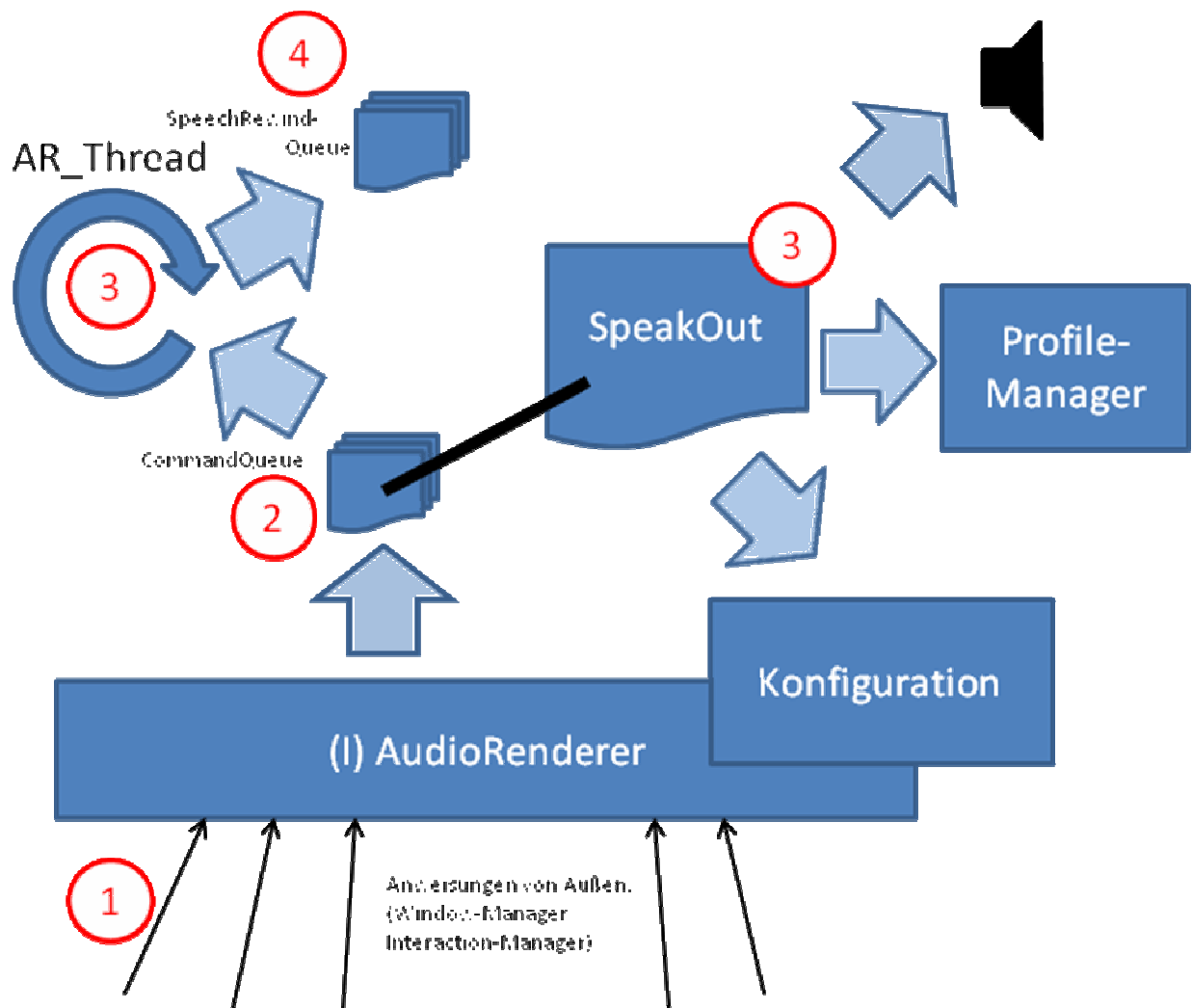


Abbildung 24 Schrittweiser Aufbau der AudioRenderer-Ausgabe: 1.) Initiierung durch den Interaction- oder WindowManager. 2.) Zwischenspeichern der Anforderung 3.) Auswerten und Vorlesen des Textes. 4) Speichern des gerade Gesagten.

Die Vorgehensweise des AudioRenderers erfolgt Schritt für Schritt nach folgendem Muster:

- 1.) Der AudioRenderer wird vom InteractionManager oder vom WindowManager aktiviert.
- 2.) Der AudioRenderer stellt die richtige Aussprache (ein SpeakOut-Objekt) zusammen und stellt dieses in die Vorleseliste (CommandQueue).
- 3.) Sobald der Lautsprecher verstummt ist, nimmt sich der AudioRendererThread (AR\_Thread) das nächste Speakout-Element aus der Liste vor. Darin erstellt das SpeakOut-Objekt einen Satz, der vorgelesen werden muss. Dies macht das SpeakOut-Objekt anhand der Konfiguration und anhand des aktuellen Profils vom ProfilManager. Das Ergebnis wird über die Lautsprecher ausgegeben.

- 4.) Das eben Gesprochene wird in der SpeechRewindQueue gespeichert, um es ggf. nochmal wiederholen lassen.

Sätze, die Widgets beschreiben, werden abhängig von Profil und Sprache zusammengebastelt.

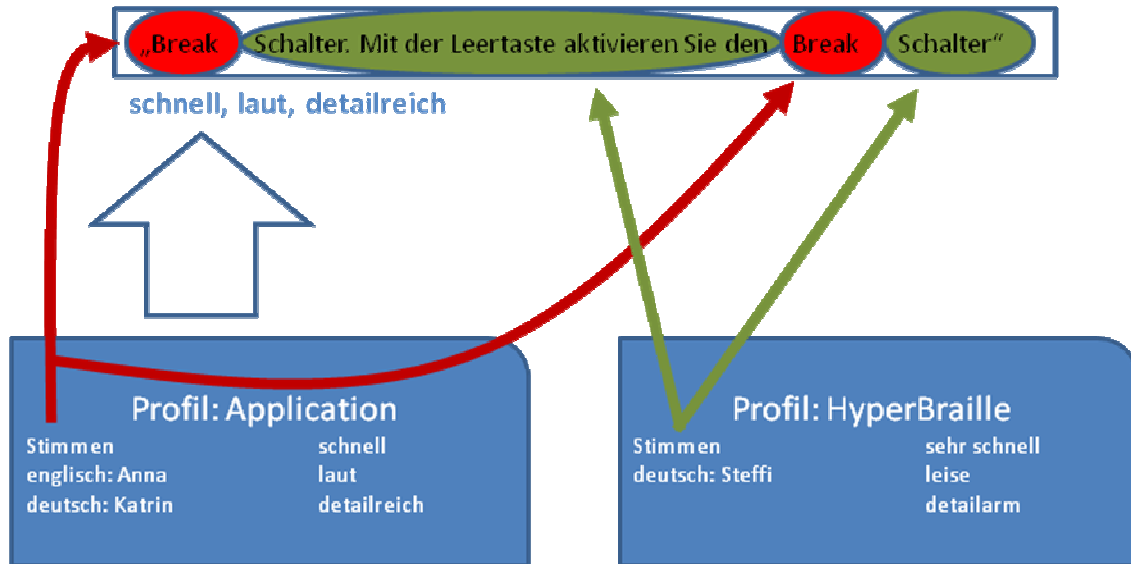


Abbildung 25: Zusammenstellung eines einfachen Satzes. Hier die Erklärung für den „Abbrechen“-Schalter in einem englischen Standardfester.

Die Abbildung 25 zeigt, wie die Profile zusammenarbeiten. Der HyperReader ist für Deutsch konfiguriert, das heißt, Hilfetexte werden in deutscher Sprache, mit einer deutschen Stimme, sehr schnell und leise ausgegeben. Werden jedoch Elemente einer Applikation erklärt, so soll die Ausgabe schnell, detailreich und laut sein. Zusätzlich soll auch die Sprache an die Programmsprache angepasst sein. In dem Beispiel wird der Abbrechen-Schalter („Break“) einer Applikation erklärt. D.h. der Inhalt des Schalters („Break“) muss in Englisch gesprochen werden, da es ein englisches Programm ist. Der Hilfetext („Schalter. Mit der Leertaste...“) muss in Deutsch gelesen werden. Der Gesamtsatz muss schnell und laut gesprochen werden.

### Ablauf der Entwicklung des AudioRenderers

Nach Start des AP 10 (Halbjahresbericht 01/2009) wurde der Renderer in den HBMonitor integriert. Es konnten die Stimmen, Lautstärke, Tonlage und Geschwindigkeit angepasst werden. Mit der Testanwendung HBMonitor konnten die Projektpartner sich bereits mit dem AudioRenderer vertraut machen.

Der nächste Schritt (Halbjahresbericht 07/2009) bestand in der Erweiterung der Anpassung an das OSM und dem Interaction Manager. So wurden für alle Widgettypen verschiedene Sprachausgaben implementiert. Zusätzlich wurde auch für Zusammenarbeit mit dem InteraktionManager gesorgt, so dass die Audioausgabe in die normale Bedienung des Anwenders integriert wurde. Beispielsweise konnte jedes Windowselement ausgesprochen werden, sobald man mit der Tab-Taste den Fokus darauf bewegte.

In der zweiten Jahreshälfte 2009 (Halbjahresbericht 01/2010) wurde die Softwarearchitektur umgestellt. Man hat festgestellt, dass die bisherige Architektur den vielen neuen Anforderungen nicht gerecht wurde. So passierte es zu oft, dass Texte nicht vollständig ausgesprochen wurden, da der WindowManager und der InteraktionManager zeitlich mehr Anfragen hatten, als in dieser Zeit überhaupt ausgegeben werden konnte. Der Blinde war nicht in der Lage zu folgen. Außerdem zeigten Benutzertests, dass die Sprachausgabe in sogenannte „Profile“ geordnet werden musste. Die neue Architektur wurde umgesetzt, und die Profile wurden implementiert.

Anfang 2010 kam das Tasturecho hinzu, ein Feature, das dem Blinden die eingegebenen Wörter wortweise wiederholt, so dass dieser weiß, ob er einen Tippfehler gemacht hat. Zusätzlich wurde der Schnellessmodus implementiert. Der Schnellessmodus arbeitet mit dem GraphicRenderer und der Stiftplatte zusammen. Sobald der Schnellessmodus aktiviert ist, spricht der AudioRenderer alle Elemente laut aus, sobald diese vom Anwender auf der Stiftplatte berührt werden. Dies hilft dem Blinden ungemein, um sich auf der Stiftplatte zu orientieren.

### **Tastatur- und Mausbehandlung - Entwicklung einer Schnittstelle**

Ein Ziel von Arbeitspaket 10 war es, die Treiberansteuerung von Windows so zu verändern, dass der Hyperreader in der Lage ist, die Eingabe von Maus und Tastatur zu manipulieren. Für einfache Behandlungen der Tastatur oder Maus ist diese Fähigkeit bereits in die Entwicklungsumgebung eingebaut, jedoch ist diese sehr beschränkt. Die Maus- und Tastaturereignisse abzufangen oder diese zu erzeugen funktioniert dabei nicht in speziellen, aber wichtigen Szenarien. Ziel hierbei ist es, die Maus- und Tastaturereignisse in **allen** Szenarien zu ermöglichen. Kritische Szenarien sind z. B. die Verwendung des LoginScreens, die Verwendung der Windows-Eingabeaufforderung, das Editieren der Systemsteuerung im abgesicherten Bereich.

Folgende Unterziele wurden in der Arbeitspaketbeschreibung definiert:

1. Abfangen aller Maus- und Tastaturereignisse
2. Ggf. Unterdrückung dieser Ereignisse
3. Ggf. Erzeugung dieser Ereignisse
4. Unterstützung verschiedener Tastaturlayouts
5. Funktionieren auch bei „Systemtechnischen Ereignissen“

Im Projektverlauf wurden zusätzlich folgende Punkte implementiert:

64-Bit Kompatibilität

### **Konzept des TastaturMausHooks**

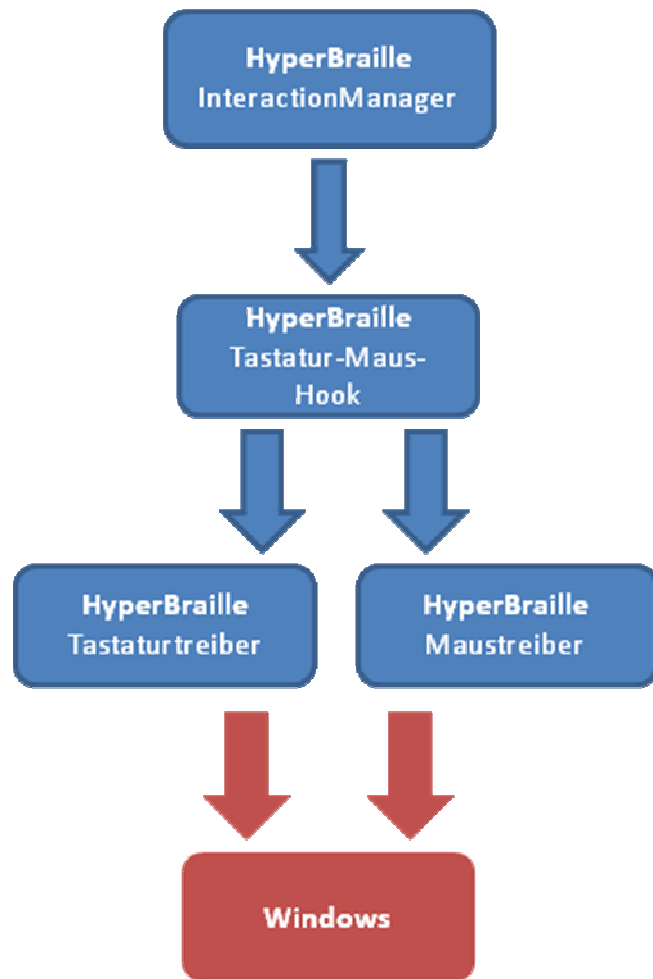
Zur Verwirklichung des TastaturMausHooks wurden im Wesentlichen zwei verschiedene Funktionen von Windows verwendet. Zum einen bietet Windows sogenannte System-Hooks an, die über interne Ereignisse informieren. Zum anderen wurden Treiber geschrieben, welche eine Maus oder eine Tastatur simulieren. Im Endeffekt wurden zwei unterschiedliche Projekte erzeugt. Das eine Projekt ermöglicht das Abfangen und Unterdrücken von Systemereignissen, das andere Projekt ermöglicht das Injizieren von Ereignissen.



**Abbildung 26: Aufbau des Tastatur- Maus- Hooks zum Abfangen von Ereignissen**

Abbildung 1 zeigt den Ablauf des TastaturMausHooks (TMH) zum Abfangen von Ereignissen. Der TMH meldet sich bei den Windows System-Hooks an. Danach gilt folgender Ablauf:

1. Ein Anwender drückt eine Taste auf der Tastatur, oder er bewegt die Maus. Daraufhin informiert der Windows-System-Hook den TMH über dieses Ereignis.
2. Der TMH bereitet das Ereignis auf und informiert den InteractionManager darüber.
3. Der InteraktionManager entscheidet entsprechend seiner Konfiguration, ob das Ereignis ausgelöst werden darf, oder ob es unterdrückt werden soll. (HyperReader-Tastatureingaben sollen nicht an Anwenderprogramme weitergeleitet werden.)
4. Entsprechend der Entscheidung aus 3 handelt der TMH und unterdrückt das Ereignis oder auch nicht.



**Abbildung 27: Ablauf des TMH beim Injizieren von Ereignissen**

Beim Injizieren von Ereignissen ist der Ablauf ein anderer.

Üblicherweise entscheidet der InteraktionManager, dass etwas injiziert werden muss. Zum Beispiel tippt der Anwender gerade mit der Brailletastatur und die Buchstaben müssen injiziert werden.

### **Ablauf der Entwicklung des TastaturMausHooks**

Nach Start des AP 10 (Halbjahresbericht 07/2008) wurde zunächst ein Versuchsbetrieb hergestellt, um Ereignisse abzufangen. Dies hat bis dahin allerdings nur ohne Administrationsrechte funktioniert.

Im zweiten Halbjahr 2008 wurden die Schwierigkeiten des ersten Halbjahres behoben. D. h., es konnten Ereignisse auch bei Prozessen mit Administrations- oder Systemrechten abgefangen werden. Zusätzlich wurden die Probleme mit einigen Spezialtasten behoben (z. B. ^ oder ~).

Im ersten Halbjahr 2009 wurde die Maus- und Tastaturinjektion hinzugefügt. Dadurch war man nun in der Lage, z. B. auf der Stiftplatte auf einen Schalter zu klicken, und dieser Schalter wurde wirklich mit der Maus auf dem Bildschirm angeklickt.



Im zweiten Halbjahr 2009 wurden die Tastaturlayouts ergänzt. D. h., man könnte nun auch schweizerische Tastaturen oder sonstige westeuropäische Tastaturen verwenden.

Im Jahr 2010 wurden keine Änderungen mehr am TMH entwickelt, es blieben lediglich Fehlerbehandlungen. Was übrig blieb, war der notwendige Test, ob der TMH auch wirklich im LoginScreen funktioniert. Dieser Test wurde im Februar 2011 bestanden.

#### **4.2.3 Tabellenverwaltung - Excel – AP15**

Blinde haben mit den bisherigen Braillezeilen enorme Schwierigkeiten, Tabellen zu bearbeiten. Das Ziel von AP15 ist es, dem Blinden eine Tabellenverwaltung zu ermöglichen. Hier in diesem Fall soll Excel verwendet werden.

Folgende Unterziele wurden in der Arbeitspaketbeschreibung definiert:

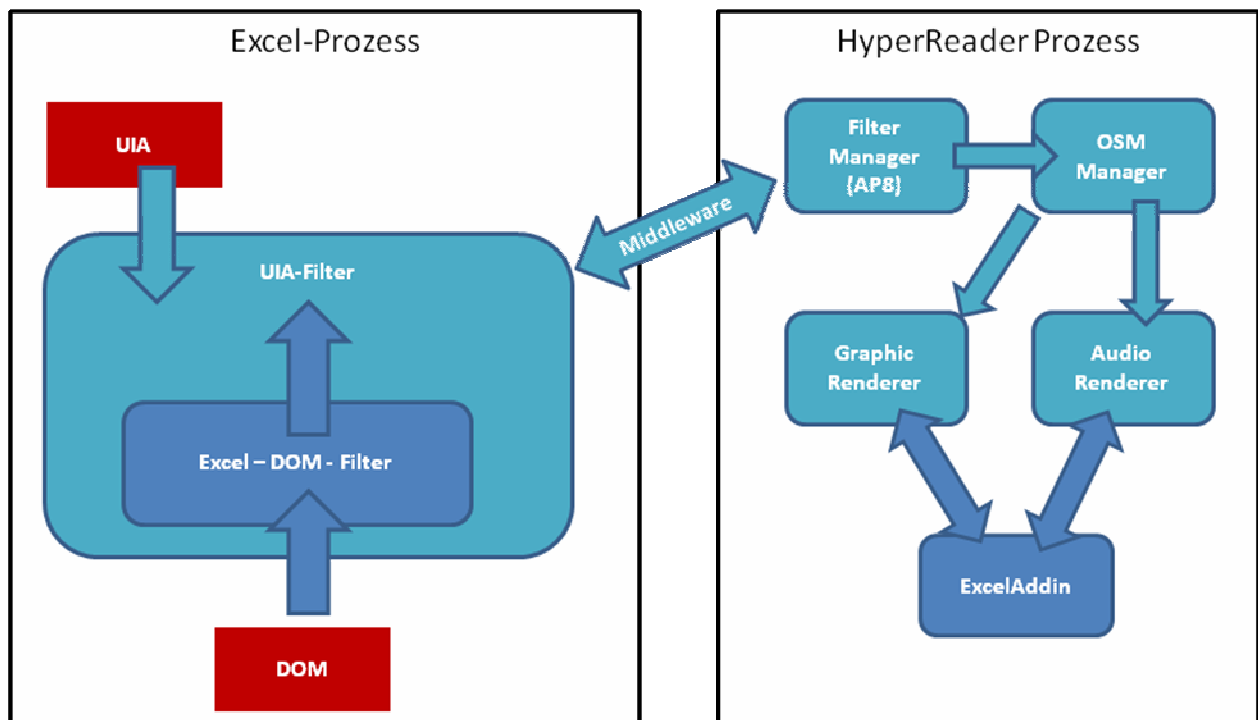
1. Entwicklung eines ExcelFilters. Der ExcelFilter soll den Inhalt des Excel-Fensters in OSM-Elemente umwandeln, die dem OSM-Manager (AP 5) übergeben werden.
2. Änderungen im Exceldokument sollen übermittelt werden.
3. Der ExcelFilter soll Befehle des FilterManagers (AP 8) verarbeiten können.

Weitere Punkte, die im Laufe des Projekts hinzukamen:

1. Auch Diagramme sollen dargestellt werden können.
2. Die allgemeine Darstellung soll speziell für Tabellen angepasst werden.

#### **Konzept des ExcelFilters**

Der ExcelFilter verwendet zur Filterung des Exceldokuments innerhalb des ExcelProzesses die sogenannte Technik „DOM“. Alle restlichen Elemente, wie z. B. das Menu, der Ribbon werden über die Technik „UIA“ gefiltert. Zusätzlich zum eigentlichen Filter gibt es noch ein Addin, welches spezielle Anpassungen im GraphicRenderer und im AudioRenderer macht.



**Abbildung 28 Einbettung von Excel in die Gesamtstruktur**

Abbildung 28 zeigt, wie Excel in das Gesamtsystem eingebettet wurde. Der ExcelFilter, welcher im Excelprocess läuft, baut auf dem vom Projekt entwickelten, allgemeinen UIA-Filter auf, der in vielen anderen Office-Filtern auch verwendet wurde. Die UIA-Technik von Windows wird verwendet, um z. B. das Menu und den Ribbon in Office zu filtern. Der ExcelFilter erweitert diesen UIA-Filter um seine eigenen DOM-spezifischen Informationen und Techniken. Der Excel-DOM-Filter ist sozusagen in den UIA-Filter eingebettet worden, um diesem Zusatzinformationen zu liefern, wie z. B. das Excelldokument mit seiner riesigen Tabelle. Der Standardablauf eines ExcelFilters ist etwa der folgende:

- 1) Der Excel-DOM-Filter erkennt eine Änderung am Excelldokument. Er wertet die Änderung aus und schickt das Ergebnis an den UIA-Filter weiter.
- 2) Dann folgt eine lange Ablaufkette, welche die Informationen weiterreicht. Der UIA-Filter übergibt die Daten über die Middleware in den HyperReaderprozess, dort an den FilterManager und dieser an den OSM-Manager. Am Ende landen die Änderungsdaten im AudioRenderer oder im GraphicRenderer.
- 3) Der GraphicRenderer oder der AudioRenderer wollen die neuen Daten darstellen, entweder als Graphik auf der Stiftplatte oder als gesprochenen Text auf der Stiftplatte. Da beide Renderer nur eine Standardimplementierung zur Darstellung haben, schaltet sich das ExcelAddin ein, um die Darstellung für Excel zu verändern. So sorgt das ExcelAddin für eine verbesserte Darstellung der Tabelle. (z. B. wird standardmäßig immer Zeilen- und Spaltennummer in Zahlen angesagt, das Addin verändert dies zu der Excelausgabe, in der die Zeilen keine Nummern sondern Buchstaben haben.)

## **Ablauf der Entwicklung des ExcelFilters**

Nach Start des AP 15 (Halbjahresbericht 07/2008) wurde die Technik DOM in Excel überprüft und getestet. Das Ergebnis war ein kleiner Filter, der einfache Dinge filtern konnte, aber nicht ins restliche HyperBraille-System integriert war.

Im zweiten Halbjahr 2008 wurde der ExcelFilter im restlichen HyperBraille-System integriert. Das heißt, der ExcelFilter wurde in den UIA-Filter eingebaut, es wurden OSM-Elemente erzeugt und versendet. Diese konnten auch schon auf der Stiftplatte dargestellt werden, jedoch nur in der Standardansicht, ohne spezielle Veränderungen. Die Exceltabelle konnte weitestgehend schon gefiltert werden, sofern es keine Diagramme oder verbundene Zellen gab.

Im ersten Halbjahr 2009 wurde die Diagrammfilterung entwickelt. So war es dann möglich, auch einfache Säulendiagramme oder Liniendiagramme zu filtern und diese auf der Stiftplatte darzustellen. Verbundene Zellen wurden von nun an auch richtig dargestellt.

Im zweiten Halbjahr 2009 fokussierte sich die Entwicklung auf die Dynamik des Filters. Das Ergebnis war, dass von nun an auch Änderungen im Dokument an den HyperReader gesandt wurden. Zu den Änderungen zählten dabei nicht nur die Änderungen, die ein Anwender direkt vornahm, sondern auch Änderungen im Dokument, die automatisch errechnet wurden (Zellen, die abhängig von anderen ihren Wert berechnen).

Im ersten Halbjahr 2010 fokussierte sich die Entwicklung darauf, viele neue Diagrammtypen zu unterstützen (z. B. das Kuchendiagramm, speziellere Balken- und Liniendiagramme).

Im zweiten Halbjahr 2010 wurde aus Performancegründen der ExcelFilter einem Redesign unterzogen, um schnellere Antwortzeiten zu erreichen. Zusätzlich fokussierte sich die Entwicklung auf die Implementierung des ExcelAddins, um Excel-Spezialansichten zu implementieren.

### **4.2.4 Softwarekomponente Signierung – AP17**

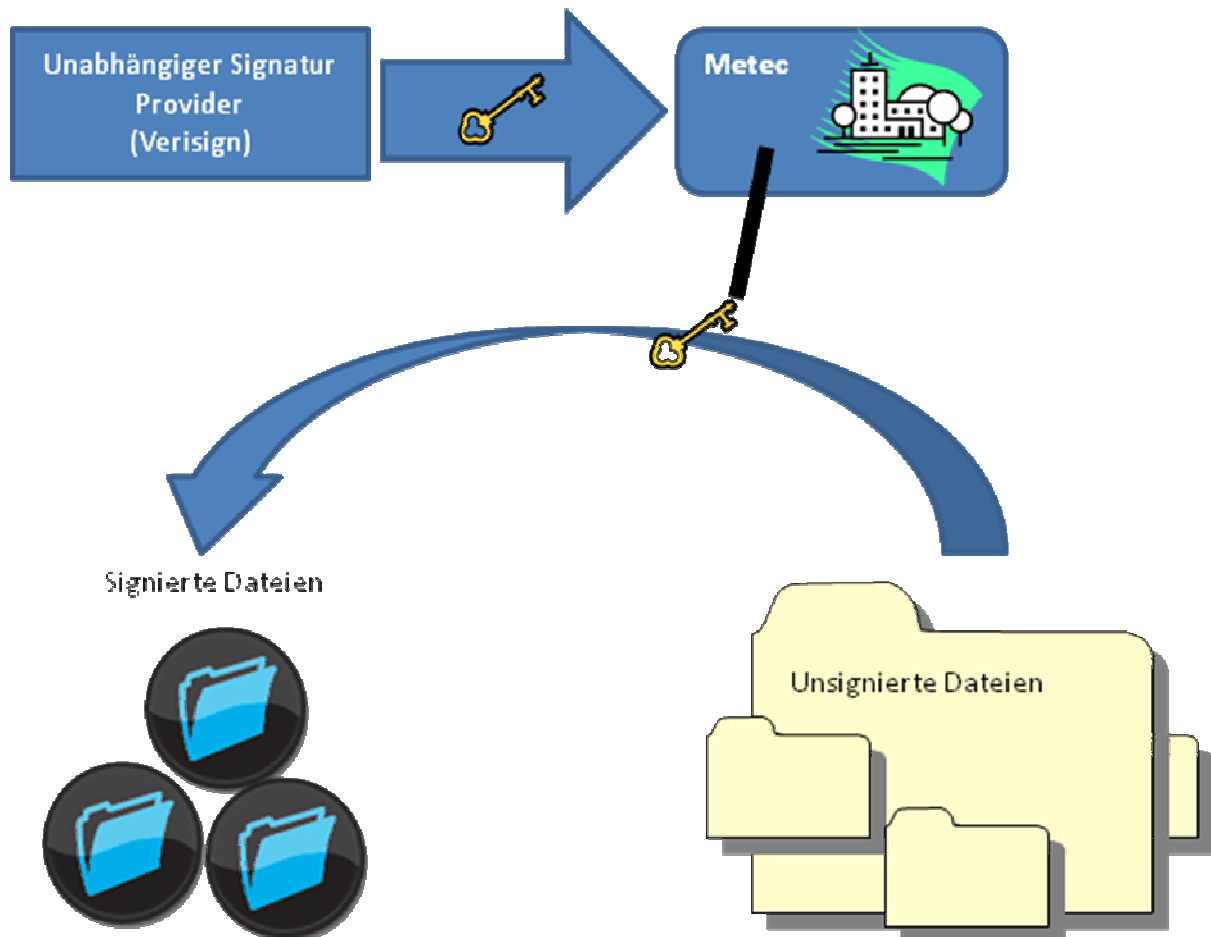
Das Arbeitspaket 17 wurde dafür ausgelegt, nicht vorhersehbare Anforderungen, die im Projekt auftreten, zu berücksichtigen. Die meisten Änderungen, die dieses Arbeitspaket in Anspruch genommen hat, sind in den Arbeitspaketen beschrieben, die das Redesign durchführen mussten (z. B. AP 10, AudioRenderer) Es gab lediglich einen Punkt, der so sehr aus dem Rahmen fällt, dass er nicht in den anderen Paketen beschrieben ist: Die Signierung.

#### **Konzept der Signierung**

Die Signierung ist ein Konzept, welches einem Windowsanwender zusätzliche Sicherheit ins System bringt. Die Signierung ermöglicht Programmentwicklern, Programme zu schreiben, die deutlich weniger lästige Sicherheitsabfragen enthalten. Die Signierung erklärt dem Anwender, **wer** ein Programm geschrieben hat. Je nach Sicherheitseinstellung, fragt Windows bei jedem systemkritischen Zugriff beim Anwender und Computeradministrator nach, ob dieser wirklich diese systemkritische Aktion durchführen will. Ohne eine Signierung würde diese Frage wirklich jedes mal

gestellt werden. Mit Signierung wird die Frage entweder gar nicht oder nur einmal bei der Installation gestellt.

HyperBraille hat einige Komponenten, die auf systemkritische Bereiche zugreifen, z. B. die Tastaturinjizierung, die Mausinjizierung, Login, Systemsteuerungsfenster, etc. So wurde es dringend notwendig, eine Signierung in das Projekt einzubauen.



**Abbildung 29: Ablauf der Signierung**

Abbildung 29 zeigt den Ablauf einer Signierung. Die Metec-AG kaufte hierfür einen Schlüssel bei einem Signatur-Provider, in diesem Fall Verisign. (Es dürfen keine Markennamen sein, es muss immer ein Firmenname sein). Mit Hilfe dieses Schlüssels kann nun jede einzelne Datei des Projekts signiert werden. Somit enthält dann jede Datei den Herstellernamen und eine Technische Absicherung, dass diese Signierung nicht von Fremden oder von einem Virus verändert werden kann. Microsoft Windows akzeptiert diese Signierung und lässt signierte Programme in fast alle Sicherheitsbereiche von Windows.

### **Ablauf der Entwicklung der Projektzusammenfassung**

Anfang 2009 wurde ersichtlich, dass eine Signierung notwendig wird, da es ein paar Elemente im HyperReader gibt, die sehr kritische Windowssystemelemente verwenden. Diese sehr kritischen Windowssystemelemente dürfen nur von signierten

Programmen verwendet werden. Es wurde uns also im zweiten Quartal 2009 beschlossen, die Signierung einzubauen. Mitte 2009 war die Implementierung dazu im vollen Gange, Ende 2009 war der HyperReader vollständig signiert, bis hin zum Setupprogramm.

Mitte 2010 musste der Schlüssel von Verisign erneuert werden, was eine kleine Projektaktualisierung zur Folge hatte.

#### **4.2.5 Kalenderverwaltung – Outlook - AP19**

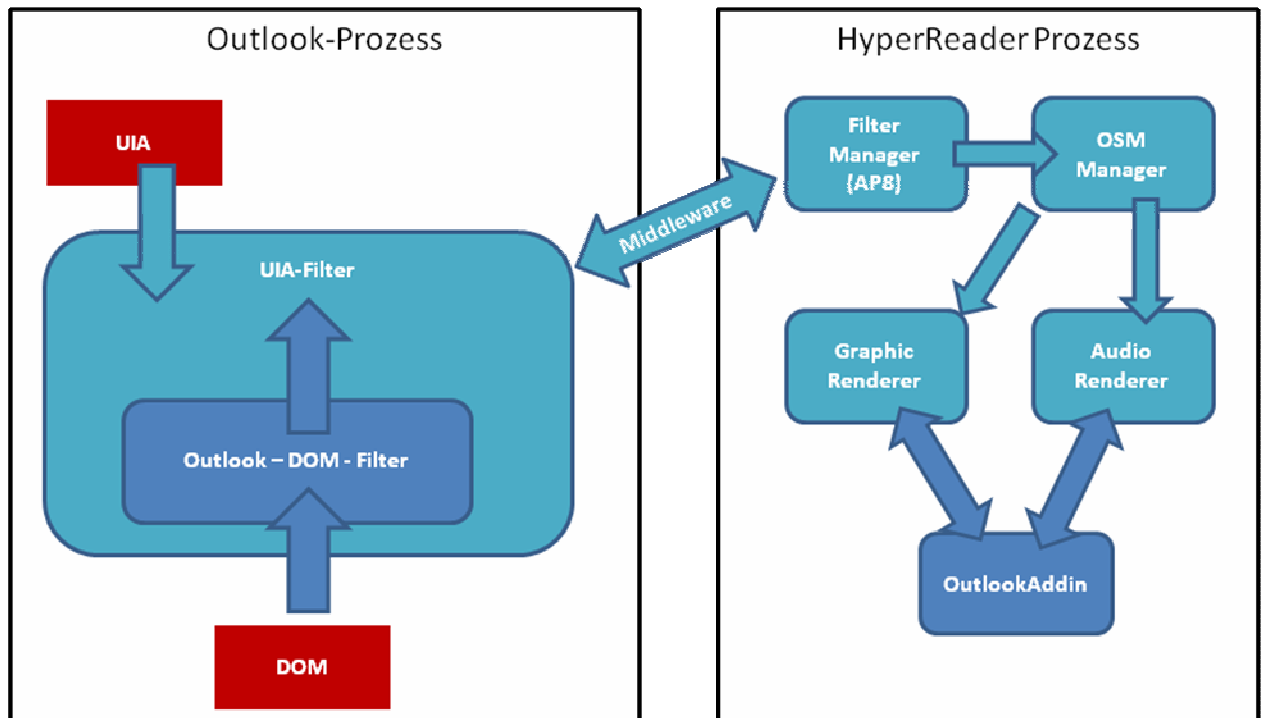
Blinde haben mit den bisherigen Braillezeilen enorme Schwierigkeiten, Kalender zu bearbeiten. Sie haben oftmals nur Terminlisten, die keinen guten Überblick über ihre Termine bieten. Das Ziel von AP19 ist es, dem Blinden das Prinzip eines Kalenders zu veranschaulichen.

Folgende Unterziele wurden in der Arbeitspaketbeschreibung definiert:

1. Entwicklung eines OutlookFilters. Der OutlookFilter soll den Inhalt des Outlook-Fensters in OSM-Elemente umwandeln, die dem OSM-Manager (AP 5) übergeben werden.
2. Übermittlung der Änderungen in Outlook-Email und Kalender
3. Verarbeitung der Befehle des FilterManagers (AP 8) durch den OutlookFilter

#### **Konzept des OutlookFilters**

Der OutlookFilter verwendet zur Filterung des Kalender und der Emails innerhalb des OutlookProzesses die sogenannte Technik „DOM“. Alle restlichen Elemente, wie z. B. das Menu, werden über die Technik „UIA“ gefiltert. Zusätzlich zum eigentlichen Filter gibt es noch ein Addin, welches spezielle Anpassungen im GraphicRenderer und im AudioRenderer macht.



**Abbildung 30: Einbettung von Outlook in die Gesamtstruktur**

Abbildung 30 zeigt, wie Outlook in das Gesamtsystem eingebettet wurde. Der OutlookFilter, welcher im Outlookprozess läuft, baut auf dem vom Projekt entwickelten, allgemeinen UIA-Filter auf, der in vielen anderen Office-Filtern auch verwendet wurde. Die UIA-Technik von Windows wird verwendet, um z. B. das Menu in Office zu filtern. Der OutlookFilter erweitert diesen UIA-Filter um seine eigenen DOM-spezifischen Informationen und Techniken. Der Outlook-DOM-Filter ist sozusagen in den UIA-Filter eingebettet worden, um diesem Zusatzinformationen zu liefern, wie z. B. den Kalender in seinen verschiedenen Formen wie Wochenansicht, Jahresansicht, etc. Der Standardablauf eines OutlookFilters ist etwa der folgende:

- 4) Der Outlook-DOM-Filter erkennt eine Änderung am Kalender/Email. Er wertet die Änderung aus und schickt das Ergebnis an den UIA-Filter weiter.
- 5) Dann folgt eine lange Ablaufkette, welche die Informationen weiterreicht. Der UIA-Filter übergibt die Daten über die Middleware in den HyperReaderprozess, dort an den FilterManager und dieser an den OSM-Manager. Am Ende landen die Änderungsdaten im AudioRenderer oder im GraphicRenderer
- 6) Der GraphicRenderer oder der AudioRenderer wollen die neuen Daten darstellen, entweder als Graphik auf der Stiftplatte oder als gesprochener Text auf der Stiftplatte. Da beide Renderer nur eine Standardimplementierung zur Darstellung haben, schaltet sich jedoch das OutlookAddin ein, um die Darstellung für Outlook zu verändern. So sorgt das Outlook-Addin für eine verbesserte Darstellung des Kalenders.

## **Ablauf der Entwicklung des OutlookFilter s**

Das AP19 fiel in den Zeitrahmen der Projektverlängerung ab April 2010. Es wurde entsprechend schnell damit begonnen. Mitte 2010 war ein Filter vorhanden, der schon den Kalender gefiltert hatte, jedoch wurde bis zu diesem Zeitpunkt der Fokus ausschließlich auf die Filterung gelegt, nicht auf die Darstellung. Ende 2010 wurde der OutlookFilter fertig. Die Filterung des Kalenders in vielen verschiedenen Ausfertigungen funktioniert. Die Darstellung des Kalenders fand Anklang bei den Blinden.

### **4.3 Zusammenfassung der Softwareentwicklung**

Die Softwareentwicklung und die dazugehörigen Tests brachten enorm viele Erkenntnisse. So wissen wir nun, in welcher Art und Weise wir einem Blinden den Computer auf einem zweidimensionalen Braille-Display präsentieren sollten. Jetzt kann ein Blinder den vollen Nutzen aus einem BrailleDisplay ziehen und deutlich effizienter als nur mit einer Braillezeile arbeiten. Es hat sich herausgestellt, dass das Arbeiten mit den vier verschiedenen Ansichtsarten sehr vorteilhaft ist.

Des Weiteren hat die Softwareentwicklung einige schwierige Knackpunkte aufgezeigt, die in der weiteren Softwareentwicklung betrachtet werden müssen. So ist beispielsweise in vielen Bereichen der Software das Betriebssystem Windows oftmals eine große Hürde gewesen, die es zu überwinden galt, da das Sicherheitssystem von Windows viele Zugriffe auf dargestellte Informationen verhindert. Probleme dieser Art gab es unter älteren Betriebssystemen noch nicht, und diesbezüglich konnten viele Erfahrungen gesammelt werden.

Alles in Allem wurde ein in sich stimmiges Softwarepaket erzeugt. Es kann einfach installiert werden, gut bedient werden und es erfüllt die Anforderungen eines Blinden. Nichtsdestotrotz ist dieses Softwarepaket noch keine Software, welche einen Qualitätstand erreicht hat, um sofort auf dem Markt verkauft werden zu können. Das Endergebnis des Projekts erfordert noch viel Nacharbeit. Noch gibt es gelegentliche Abstürze und Speicherlöcher. Für die einzelnen Programme, wie z. B. Office/Word, sind zwar gute Filter geschrieben worden, jedoch sind nicht alle Möglichkeiten dem Blinden zugänglich gemacht worden. Im Normalfall beschränkten sich die Filter auf Features, die regelmäßig gebraucht werden, oder solche, die für Forschung und neue Erkenntnisse interessant waren. So ist in Word die Darstellung des Dokuments geregelt, da man dies immer braucht, und die Rechtschreibprüfung wurde implementiert, da das für einen Blinden durchaus etwas Neues ist, aber Tabellen innerhalb von Word können noch nicht dargestellt werden. Das Thema „Tabellen“ wurde jedoch in Excel behandelt und eine Einbindung in Word fehlt noch.

## 5. Notwendigkeit und Angemessenheit der geleisteten Arbeit

Derzeit technischer Stand für Arbeitsplatzausrüstungen für blinde Menschen sind Braillezeilen mit Sprachausgabe. Diese Geräte sind weitgehend limitiert auf Textdarstellungen. Damit lassen sich jedoch Daten von aktuellen Anwendungen wie Word, Excel oder auch Internet Explorer nur sehr begrenzt darstellen. Dies bedeutet immer mehr die Ausgrenzung von blinden Menschen. Mit der Entwicklung von einem Flächendisplay wird dies wesentlich vermindert, und Blinde können selbständig diese Programme weitgehend vollwertig und allein nutzen. Dadurch steigt die Möglichkeit, am Arbeitsleben teilzunehmen, wieder ganz deutlich. Ein geeignetes Gerät zur Darstellung von graphischen oder flächigen taktilen Darstellungen bestand bisher nicht. Hier war es notwendig, im Projekt ein entsprechendes usergerechtes Gerät zu entwickeln. Das in HyperBraille entstandene Flächendisplay wurde nach Maßgabe der User und Softwareexperten gestaltet. Es flossen die Ergebnisse der Usertests genauso ein wie die Forderung der Softwareexperten nach einer höheren Auflösung der Touchsensorik. Damit entstand im Projekt eine Hardware, die zumindest im Aufbau und Erscheinungsbild schon nahe zu einem Seriengerät ist.

Parallel hierzu wurde eine Software, die dem heutigen Stand der Technik entspricht, entwickelt. Die taktile Ausgabe musste sowohl an die Bedürfnisse der blinden Menschen als auch an die Möglichkeiten der Hardware angepasst werden. So ist z. B. die Auflösung der Flächendisplays mit 7200 Punkten immer noch wesentlich geringer als in Bildschirmdarstellungen. Dies hat zur Folge, dass ganz anders gezoomt, gescrollt und auch dargestellt werden muss. Das flächige Display bietet viele neue Möglichkeiten, wie Daten dargestellt werden können. Es war notwendig herauszufinden, in welcher Art und Weise Daten dargestellt werden können und mit welcher Effizienz und Ergonomie verschiedene Ansichten praktisch verwendet werden können. Die besten Ansichtsarten wurden implementiert, um einen sinnvollen Gebrauch des Flächendisplays zu gewährleisten. Zum schnellen und ergonomischen Arbeiten muss zwischen verschiedenen Ansichten gewechselt werden. Z. B. müssen Texte an entsprechender Stelle sowohl in Brailleschrift als auch umschaltbar in Schwarzschrift dargestellt werden. Sprachausgabe muss die Befehlseingabe und Navigation unterstützen. Dies bedingt eine komplett neue Ausgabesoftware, damit die oben genannten Punkte optimal für blinde User aufbereitet werden. Damit wird die Ausgabe der Standardprogramme (wie Excel oder PowerPoint) so aufbereitet, dass blinde User damit graphisch arbeiten können. In den Arbeitspaketen wurde dies konsequent und zielgerichtet verwirklicht.

Im Zusammenspiel von Hard- und Software entstand ein neuartiges taktiler Ausgabegerät, das blinden Menschen in Zukunft die autarke Benutzung von Computern ermöglicht und damit die Aussichten auf einen Arbeitsplatz ganz deutlich verbessert.



## 6. Fortschritt bei anderen Stellen

Parallel zu den Projektarbeiten wurde eine „Technologiestudie zu Brailledisplays“ angestellt. Hier wurden alternative Möglichkeiten der taktilen Ausgabe und Fortschritt bei anderen Stellen recherchiert. Es wurde der Stand der Technik bei klassischen Aktuatoren wie elektromagnetische Motoren, Spulen und Fluidik, bei der Pneumatik und Thermopneumatik untersucht. Ein weiteres Feld bestand bei den Festkörperaktoren, wie Ferroelektrika, Polymere, magnetostruktive Materialien, elektrorheologische Flüssigkeiten und Formgedächtnislegierungen. Zusammenfassend wurde der Fortschritt bzw. Entwicklungsstand von neuartigen taktilen Displays durchaus mit stark steigendem Interesse in den letzten Jahren bezeichnet, jedoch zeigen nur wenige Systeme Aussicht auf Erfolg. Die alternativen Systeme stehen aber noch weit vor der praktischen Verwirklichung, so dass die piezobetriebenen HyperBraille-Displays die derzeit einzig verwirklichtbaren darstellen.

## **7. Anwendungspotentiale und Verwertung der Ergebnisse**

Die Anwendungspotentiale gliedern sich prinzipiell in 2 Bereiche. Zum einen in den Einsatzbereich und zum zweiten in die Zielgruppen. Die Verwertung hängt damit an den entsprechenden wirtschaftlichen Aussichten der einzelnen Anwendungen.

### **7.1 Einsatzbereiche**

#### **7.1.1 Beruf, Schule, Studium**

Die ersten Einsatzbereiche der HyperBraille-Flächendisplays in Verbindung mit der neu entwickelten Software werden kosten- und nutzenbedingt zuerst Arbeitsplätze und Forschungsprojekte an Universitäten sein. In Bezug auf die neue Gestaltung eines Blindenarbeitsplatzes und damit der verbesserten Integration und höheren Leitungsfähigkeit bzw. Einsatzmöglichkeit von blinden Menschen im Betrieb wird diese Lösung sich am schnellsten wirtschaftlich für den Betreiber auszahlen.

Der Einsatz in Schulen und Universitäten zur besseren Ausbildung von blinden Menschen sollte parallel starten. Hier wird mit dem System der Grundstein für den beruflichen Einstieg in Berufsgruppen, die bisher von Blinden kaum besetzt sind, gelegt. So sind hier Tätigkeiten z. B. in der EDV, IT und im kaufmännischen Bereich denkbar.

#### **7.1.2 Öffentliche Gebäude, Museen, Institute**

In öffentlichen Gebäuden Bibliotheken, Museen etc. können die Displays zur Orientierung und Beschreibung der Räumlichkeiten sowie zur Erklärung von Sachverhalten eingesetzt werden. Dies kann zentral an einer Stelle ähnlich einem öffentlichen Internetplatz oder auch dezentral direkt an einer Örtlichkeit, an der die Information benötigt wird, erfolgen. Damit kann z. B. eine Bibliothek weitgehend barrierefrei für blinde Menschen gestaltet werden. Hier sind neben der Orientierung im Gebäude auch die Werke selber darstellbar in Titel, Abstracts, Graphiken etc.

#### **7.1.3 Forschungsprojekte**

Der Einsatz in Forschungsprojekten ergibt ganz neue Ansätze und Aussichten. Zum einen sind Anwendungen in Projekten wie „Haptimap“ zur Darstellung von Karten, Wegen, Gebäuden etc. denkbar und sinnvoll, zum anderen können durch die Weiterentwicklung der Software in Projekten, z. B. in der optimalen Darstellung und Interaktion von Programmen der Mathematik, Chemie oder Physik, später in Lehre und Beruf, stark erweiterte Berufsbilder von blinden Menschen besetzt werden. Hier schließt sich der Kreis zur Arbeitsplatzanwendung.

#### **7.1.4 Privater Bereich**

Sofern es in den nächsten Jahren gelingt, die Hardware deutlich kostengünstiger zu produzieren, ist auch der Einsatz im privaten Bereich zum Surfen im Internet, Lesen von Büchern, E-Mail Bearbeitung etc. möglich. Hier bedarf es aber noch einiger Anstrengung, um diese Preisziele zu erreichen.

## 7.1.5 Sonstige Bereiche

Abgewandelte Hardwaresysteme mit spezieller Software sind im Bereich der Haussteuerungen für private Haushalte aber auch zur barrierefreien Steuerung von öffentlichen Gebäuden, z. B. Blindenschulen, einsetzbar. Zentral gesteuert werden hiermit Heizung, Lüftung, Türöffner, Licht etc. Auch Sicherheitssysteme lassen sich hierbei einbinden. Seit einiger Zeit besteht hier schon eine Zusammenarbeit mit einem Partnern des Projektes „Digital Strom“.

## 7.2 Zielgruppen

Entsprechend den oben genannten Einsatzgebieten ergeben sich unterschiedliche Zielgruppen.

### **7.2.1 Zielgruppe der blinden Nutzer, die 2011 bereits im Beruf stehen**

Diese werden sich hauptsächlich in den Einsatzbereichen Beruf und auch privater Bereich wieder finden.

Diese Zielgruppe besteht aus dem Anteil blinder Menschen, die aus Neigung oder Interesse oder wegen beruflicher Anforderungen HyperBraille als eine wertvolle Hilfe für die Verbesserung ihrer Arbeits- und Kommunikationsbedingungen kennen lernen. Diese Zielgruppe wird sich in den ersten drei Jahren ab 2011 vor allem in Deutschland finden. Erst durch „Multiplikator-Effekte“ (weitere Entwicklungsarbeiten von den Hochschulpartnern, Vorträge mit Demonstrationen auf nationalen und internationalen Fachkongressen, Steigerung der Bekanntheit in den Blindenverbänden) wird sich diese Zielgruppe in späteren Jahren auch auf internationalem Gebiet ausdehnen.

Geschätzter ansprechbarer Altersbereich :	25–40
Geschätzte Zielgruppe in BRD:	5.000 blinde Menschen
Tatsächlich nachhaltig Interessierte 2012 - 2016:	500 blinde Nutzer
Erwartete Umsätze nur BRD, 2012 - 2016:	150 HyperBraille-Systeme
Erwartete Umsätze international, 2014 - 2018:	350 HyperBraille-Systeme

### **7.2.2 Zielgruppe der zukünftigen blinden Nutzer, die bereits zum Abitur mit HyperBraille-System gearbeitet haben und über ein Studium mit HyperBraille-System in eine Berufsausübung mit HyperBraille-System „hineinwachsen“.**

Diese Zielgruppe wird erst nach dem Einsatz von HyperBraille-Systemen in der Schule und im Studium diese dann am Arbeitsplatz entsprechend einsetzen.

Beginnend eventuell beim Partner Blista, später parallel an weiteren Blindenschulen, die mit Abitur oder anderen anerkannten Schulabschlüssen blinde Schüler auf weiter fortführende Studien- oder Berufsausbildungsgänge vorbereiten, soll das HyperBraille-System in das Curriculum eingeführt werden und u. a. als Hilfe in der Ausbildung, besonders in naturwissenschaftlichen Fächern, dienen.

Erwarteter Bedarf BRD, 2012- 2016 :	30 HyperBraille-Systeme
-------------------------------------	-------------------------

Die Entwicklung bzw. Erweiterung des HyperBraille-Systems um die anwenderorientierte Integration der fachbezogenen Notationen und Softwareanforderungen (Fachgebiete Chemie, Physik, Mathematik, Elektrotechnik, Maschinenbau, CAD-Schnittstellen), die für ein integratives Studium und selbständige Ausübung eines entsprechend qualifizierten Fachberufs durch blinde Menschen ermöglicht wird.

Erwarteter Bedarf Entwicklungsprojekt, 2012 - 2014: 10 HyperBraille-Systeme

Erwarteter Bedarf nach Abschluss, 2015 - 2018: 350 HyperBraille-Systeme

### **7.2.3 Zielgruppe aller blinden Menschen unabhängig von der Ausbildung oder dem Alter.**

Diese Zielgruppe wird sich mehrheitlich in HyperBraille-Anwendungen und Systemen wieder finden, in denen keine besondere Ausbildung bzw. Einarbeitung nötig ist. Dies bezieht sich im Wesentlichen auf Systeme in öffentlichen Gebäuden, Museen und Instituten zur Information und Navigation sowie evtl. in Bereichen der Haussteuerungen.

Erwarteter Bedarf Entwicklungsprojekt, 2012 - 2014: 30 HyperBraille-Systeme

### **7.2.4 Zielgruppe Forschungsprojekte**

In diesem Bereich sind vor allem Weiterentwicklungen der Software sowie neue Einsatzmöglichkeiten interessant. Eine Abschätzung fällt hier sehr schwer, jedoch zeichnet sich ab, dass dies zu den ersten Einsatzbereichen der HyperBraille-Systeme führt.

Erwarteter Bedarf Entwicklungsprojekt, 2012 - 2014: 30 HyperBraille-Systeme

## **7.3 Verwertung**

Das Projekt hat die Voraussetzungen geschaffen, bis zum Jahresende 2011 die Soft- und Hardware des HyperBraille-Systems soweit fertig zu stellen, dass eine erste Vorserie gestartet werden kann. Damit ist geplant, ab Anfang 2012 ein HyperBraille-System anzubieten, das aus dem Display 7200 und einer lauffähigen Grund-Software für die gängigen Officeprogramme und Internetbrowser ausgestattet ist. Mit dieser ersten Version sind die Einsatzbereiche Beruf, Schule und Forschungsprojekte realisierbar. Das HyperBraille-System wird fast nur aus in Deutschland hergestellten Komponenten aufgebaut und am Standort Stuttgart fertig montiert.

Für die Bereiche Studium, Öffentliche Gebäude, Haussteuerungen etc. sind Soft und Hardwareanpassungen nötig. Im Bereich der Hardware ist dies durch die modulare Aufbauweise ohne größere Aufwendungen möglich. Im Softwarebereich hängt dies stark von den einzelnen Anforderungen ab. So ist im Bereich der Haussteuerungen der Aufwand absehbar, im Bereich Studium sind hier noch größere Anstrengungen und evtl. Projekte nötig.

Unter „7.2 Zielgruppen“ ist der erwartete Bedarf pro Zielgruppe schon kurz aufgezeichnet.

Die Verwertungsaktivitäten haben bereits begonnen. So wurde das System z. B. schon das 2. Mal auf der weltgrößten Blinden- und Sehbehinderten Messe SightCity Frankfurt vorgestellt. Weitere Messen wie die Braille21 in Leipzig und Bekanntmachung durch Radiobeträge folgen. Diverse Anfragen von laufenden und neu zu beantragenden Projekten sind ebenfalls schon eingetroffen. Kontakte zum Aufbau von Haussteuerungen im Rahmen des Digital Strom Projektes laufen ebenfalls schon seit geraumer Zeit. In Bezug auf die Aktivitäten in Schule und Beruf sind noch vermehrt Anstrengungen nötig.

Aufgrund der relativ teuren Piezotechnologie sind die umsetzbaren Stückzahlen begrenzt. Hier muss in Zukunft an alternativen Lösungen gearbeitet und geforscht werden. Diese Entwicklung bedeutet, sofern sie nicht mit Projektpartnern bzw. mit der Metec-AG stattfinden, das derzeit größte Risiko in Bezug auf Preisverfall und Stückzahlen. Hier sind die Märkte bzw. die Entwicklungen sorgfältig zu beobachten.

#### **7.4 Gebrauchswert einer HyperBraille-Einheit**

Der Gebrauchswert von HyperBraille ist durch die Erschließung von Grafik bei den Officeprogrammen und erheblich verbessertem Internetzugang ein Quantensprung gegenüber dem Gebrauchswert der Zeilentechnik. Das System kombiniert optimal die Vorteile der Brailleschrift mit der Darstellung von Graphiken, Layouts und auch interaktiven Windows-Desktop. Darstellungen von mathematischen oder chemischen Formeln lassen sich hier erstmals taktil und dynamisch darstellen. Damit gelingt im Blindenbereich der Übergang in das Windows- und Graphikzeitalter. Durch die Interaktion kann das Display ähnlich einem Touchscreen bedient und gelesen werden.

Die systematische Marktdurchdringung und damit wahre Würdigung des Gebrauchswertes des HyperBraille-Systems wird mit der systematischen, bereits in der Schule beginnenden Ausbildung der nachwachsenden Generationen von blinden Schülern und Studenten eintreten.

## 8. Abbildungsverzeichnis

Abbildung 1	Stiftplatte1 aus überarbeiteten BrailleDis-Modulen .....	9
Abbildung 2	überarbeitetes BrailleDis-Modul komplett .....	9
Abbildung 3	Modulaufbau Teilschritte.....	10
Abbildung 4	Aufbau Sensor und Referenzstruktur.....	11
Abbildung 5	BrailleDis-Modul mit Chips und Steckerfahne.....	11
Abbildung 6	BrailleDisTestaufbau mit 48 Modulen .....	12
Abbildung 7	HyperBrailleDisplay 7200 komplett .....	14
Abbildung 8	10 Punkte Modul komplett aufgebaut .....	15
Abbildung 9	Spritzgussteil Modulgehäuse für LDS Technik .....	15
Abbildung 10	schematische Darstellung der LDS Technik .....	16
Abbildung 11	Sensor- und Leiterbahnaufbau im LDS-Verfahren.....	17
Abbildung 12	Montagevorrichtung für die Piezobiegwandler .....	18
Abbildung 13	Sensorstrukturen im Vergleich BrailleDis und HyperBraille-Modul .....	19
Abbildung 14	Referenzkondensator beim BrailleDis-Modul.....	19
Abbildung 15	Anschlussleitungen im Vergleich .....	19
Abbildung 16	HyperBraille-Display Testaufbau mit 48 Modulen .....	21
Abbildung 17	HyperBraille-Display in Betrieb .....	22
Abbildung 18	Aufbau des HyperReaders .....	24
Abbildung 19	Screenshot einer Installation .....	25
Abbildung 20	Ansicht des HBMonitors; das Bild zeigt den Zustand des BrailleDisplays, um den Desktop in der Überblicksansicht darzustellen.....	27
Abbildung 21	Überblick über Softwaremodule und deren Zuordnung zu verschiedenen Arbeitspaketen .....	28
Abbildung 22	Aufbau der Stiftplattenansteuerung .....	30
Abbildung 23	Aufbereitungspipeline für die Audioausgabe .....	34

Abbildung 24 Schrittweiser Aufbau der AudioRenderer-Ausgabe: 1.) Initiierung durch den Interaction- oder WindowManager. 2.) Zwischenspeichern der Anforderung. 3.) Auswerten und Vorlesen des Textes. 4) Speichern des gerade Gesagten. ....	36
Abbildung 25 Zusammenstellung eines einfachen Satzes. Hier die Erklärung für den „Abbrechen“-Schalten in einem englischen Standardfester.....	37
Abbildung 26 Aufbau des Tastatur-Maus-Hooks zum Abfangen von Ereignissen....	39
Abbildung 27 Ablauf des TMH beim Injizieren von Ereignissen.....	40
Abbildung 28 Einbettung von Excel in die Gesamtstruktur .....	42
Abbildung 29 Ablauf der Signierung .....	44
Abbildung 30 Einbettung von Outlook in die Gesamtstruktur .....	46

## 9. Erfindungen/ Schutzrechte

- 1.) Patent eingereicht am 08.02.2010 mit der Bezeichnung „Taktiler Display-Deckplatte“ mit der Nummer 10 2010 005 933 1.
- 2.) Bezeichnung „Taktiler Display – Zungenbefestigung“ ist am 13.11.2008 unter dem Aktenzeichen DE 10 2007 022 056 A1 offen gelegt.
- 3.) Bezeichnung „Taktiler Display – Kontrollbox“ ist am 31.10.2007 unter dem Aktenzeichen DE 10 2006 020 568 A1 offen gelegt.
- 4.) Bezeichnung „Taktiler Display – Touch-Widerstand“ ist am 12.04.2007 unter dem Aktenzeichen DE 10 2005 047 837 A1 offen gelegt.



## 10. Veröffentlichungen / Literatur

- WWW
  - [www.HyperBraille.de](http://www.HyperBraille.de) (.eu und .com) ist realisiert (führte zu diversen Anfragen)
  - Markenschutz erhalten, Marke „HyperBraille“ wurde geschützt.
  
- Tagungen
  - International Conference on Computers for Handicapped Persons 2008, Linz
  - Tactile Graphics 2008, Birmingham
  - HCI International 2009, San Diego (TUD)
  - PETRA 2009, Corfu (Uni Potsdam)
  - International Conference on Computers for Handicapped Persons 2010, Linz (TUD)
  - BFG-IT (2010), Tagung des Fachausschusses für IT-Systeme (FIT) in der Blinden- und Sehbehindertenselbsthilfe
  - OUTPUT Dresden (2011)
  - „Lange Nacht der Wissenschaft“, Dresden 2011
  
- Veröffentlichungen
  - Albert, K (2008) Konzeption und Entwicklung eines taktilen User Interfaces, Institut für angewandte Informatik, TU Dresden, Großer Beleg
  - Gutschmidt, R.; Schiewe, M.; Zinke, F.; Jürgensen, H. (2010): Haptic Emulation of Games: Haptic Sudoku for the Blind. In: 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA). Samos.
  - Köhlmann, W.; Nadig, O.; Schiewe, M.; Weber, G. (2009): What You Feel is What You Get: Mapping GUIs on Planar Tactile Displays , HCI International 2009.
  - Köhlmann, W.; Zinke, F.; Schiewe, M.; Jürgensen, H. (2010): User Interface Filter for Two-Dimensional Haptic Interaction. In: Proceedings of the 11th International Conference on Computers Helping People with Special Needs (ICCHP), Wien.
  - Kraus, M.; Völkel, Th.; Weber, G. (2008): An Off-Screen Model for Tactile Graphical User Interfaces , International Conference on Computers Helping People with Special Needs (ICCHP) 2008, 835–842.
  - Miao, M., Köhlmann, W., Schiewe, M., Weber, G. (2009): Tactile Paper Prototyping with Blind Subjects. In Haptic and Audio Interaction Design (HAID), Dresden, LNCS 0302-9743, Springer, 2009, 81-90.

- Mechmet Chiousemoglou, Helmut Jürgensen (2011): Setting the Table for the Blind. In: Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA), Kreta.
- Pfützner, F. (2008) Erkennung von Gesten innerhalb taktiler Interaktion, Großer Beleg, Institut für angewandte Informatik, TU Dresden.
- Prescher, D. (2009) Ein taktiler Fenstersystem mit Multitouch-Bedienung, Diplomarbeit, Institut für Angewandte Informatik, TU Dresden.
- Prescher, D.; Nadig, O.; Weber, G. (2010a): Reading Braille and Tactile Ink-print on a Planar Tactile Display. In: K. Miesenberger et al. (Eds.): Proceedings of the International Conference on Computers Helping People With Special Needs (ICCHP'10), LNCS 6180, 482-489.
- Prescher, D.; Weber, G.; Spindler, M. (2010b): A Tactile Windowing System for Blind Users, In Proceedings of the 10th international ACM SIGACCESS Conference on Computers and Accessibility (Orlando, Florida, USA, October 25–27, 2010). Assets '10. ACM, New York, NY.
- Schmidt M.; Weber, G. (2010): Enhancing Single Stroke Gesture Classifiers to Multitouch Support. In: K. Miesenberger et al. (Eds.): Proceedings of the International Conference on Computers Helping People With Special Needs (ICCHP'10), LNCS 6180, 490-497.
- Schmidt, M.; Gerhard Weber, G. (2009): Multitouch Haptic Interaction. In Stephanidis, Constantine (ed.): Universal Access in Human-Computer Interaction. Applications and Services 5th International Conference, UAHCI 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009, LNCS Volume 5615/2009, Springer, 574-582.
- Spindler, M.; Kraus, M.; Weber, G. (2010): A Graphical Tactile Screen-Explorer. In: K. Miesenberger et al. (Eds.): Proceedings of the International Conference on Computers Helping People With Special Needs (ICCHP'10), LNCS 6180, 474-481.
- Sturm, I.; Schiewe, I.; Köhlmann, W.; Jürgensen, H. (2009): Communicating through Gestures without Visual Feedback, International conference on Pervasive technologies for assistive environment (PETRA 09).
- Taras, C.; Raschke, M.; Schlegel, T; Ertl, T; Prescher, D.; Weber, G. (2010): Improving Screen Magnification Using the HyperBraille Multiview Windowing Technique. In: K. Miesenberger et al. (Eds.): Proceedings of the International Conference on Computers Helping People With Special Needs (ICCHP'10), LNCS 6180, 506-512.
- Taras, C.; Raschke, M.; Schlegel, Th.; Ertl, Th. (2009): Running Graphical Desktop Applications on Tactile Graphics Displays Made Easy. In: Proceedings of 3rd International Conference on Software Development for Enhancing Accessibility and Fighting Info-exclusion (DSAI 2010), 25-26 November 2010, Oxford, United Kingdom, UTAD ISBN:987-972-669-994-1, 141-147.
- Taras, C.; Ertl, Th. (2009): Interaction with Colored Graphical Representations on Braille Devices, HCI International 2009.
- van Erp, J. B. F.; Kyung, K.; Kassner, S.; Carter, J.; Brewster, S.A.; Weber, G.; Andrew, I. (2010): Setting the Standards for Haptic and Tactile Interactions: ISO's Work. EuroHaptics (2) 2010: 353-358.
- Völkel, Th.; Weber, G.; Baumann, U. (2008): Tactile Graphics Revised: The Novel BrailleDis 9000 Pin-Matrix Device with Multitouch Input , International Conference on Computers Helping People with Special Needs (ICCHP), 2008, 865–872.

- Weber, G. (2010): Designing Haptic Interaction for a Collaborative World (ICCHP Keynote). In: K. Miesenberger et al. (Eds.): Proceedings of the International Conference on Computers Helping People With Special Needs (ICCHP'10), LNCS 6180, 431-438.
- Zeng, L. and Weber G. (2009). "Interactive Haptic Map for the Visually Impaired", In Proc. 4th International Haptic and Audio Interaction Design Workshop, Vol II, Dresden, Technische Universität Dresden, 2009, 16-17.
- Zeng, L.; Weber, G. (2010): Audio-Haptic Browser for a Geographical Information System . In: K. Miesenberger et al. (Eds.): Proceedings of the International Conference on Computers Helping People With Special Needs (ICCHP'10), LNCS 6180, 466-473.
- Beiträge zu Journals in Vorbereitung
  - Transactions on Accessible Computing, ACM
  - Universal Access in the Information Society, Springer
- Messen
  - SightCity 2009, Frankfurt
  - SightCity 2010, Frankfurt
  - SightCity 2011, Frankfurt
- Presse
  - Magnus.de, Mai 2009 (<http://www.magnus.de/news/HyperBraille-grafikfaehiges-display-fuer-sehbehinderte-183671.html>)
  - Informationsdienst Wissenschaft, idw Mai 2009 (<http://www.idw-online.de/pages/de/news315219>)
  - Connect (<http://www.connect.de/news/notebook-fuer-sehbehinderte-382899.html>)
  - Golem.de, Mai 2009 (<http://www.connect.de/news/notebook-fuer-sehbehinderte-382899.html>)
  - Deutschlandfunk, Mai 2009 (<http://www.dradio.de/dlf/sendungen/forschak/965159/>)
  - Technology Review Juli 2009 (S. 14): Display für Blinde
  - Saxxess.com (2009)
  - Heise.de, 2009 (<http://www.heise.de/newsticker/meldung/HyperBraille-entwickelt-Flaechendisplay-fuer-Blinde-und-Sehbehinderte-218790.html>)
  - Spiegel Online, 2009 (<http://www.spiegel.de/netzwelt/tech/0,1518,625005,00.html>)
  - Märkische Allgemeine
  - Potsdamer Universitätszeitung

- rbb, Sendung „OZON“; 3Sat, Sendung „nano“ ([http://www.rbb-online.de/ozon/archiv/ozon\\_vom\\_12\\_10\\_2009/die\\_erben\\_von\\_braille.html](http://www.rbb-online.de/ozon/archiv/ozon_vom_12_10_2009/die_erben_von_braille.html))
- ORF (2010)
- DresdenEins, 2011 (TV, [http://www.dresdeneins.tv/service/TUWissenschaft\\_I\\_\\_PC\\_Grafiken\\_blind\\_verstehen-758.html](http://www.dresdeneins.tv/service/TUWissenschaft_I__PC_Grafiken_blind_verstehen-758.html))
- Schattenblick.de, 2011 (<http://www.schattenblick.de/infopool/panchwitz/presse/ppmes166.html>)
- zdNet.de, 2011 (<http://www.zdnet.de/news/41552642/HyperBraille-laptop-fuer-blinde-geht-in-serie.htm>)
- golem.de, 2011 (<http://www.golem.de/1105/83273.html>)
- CampusRadio Bielefeld (2011)
  
- Geplante Tagungen
  - Ausstellung zur 100-Jahrfeier des Blinden- und Sehbehindertenverbandes Württemberg e. V., Bezirksgruppe Esslingen.
  - Präsentation zur „Montags Demonstration“ am Karlsruher Institut für Technologie, Studienzentrum für Sehgeschädigte.
  - Weltkongress Braille 21 (2011), Leipzig