

Web & Software Accessibility

Java Applets, Software & Co

Gerhard Nussbaum



Überblick

- Java Applets & Accessibility
- Software Accessibility
- Richtlinien
- (Negativ) Beispiele

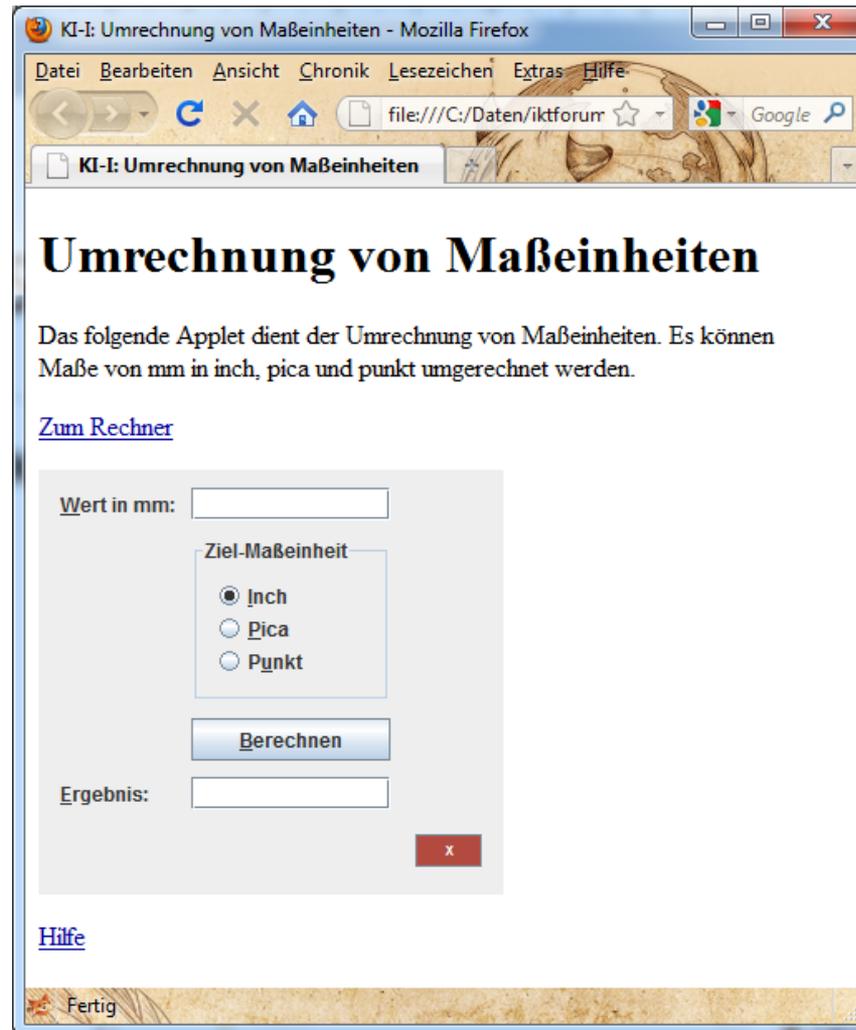


Java Applets

- Computerprogramm innerhalb einer Webseite
- Läuft komplett am Client
- Auch mit eigenem GUI
 - Java AWT
 - Java Swing
- Voller Funktionsumfang aus der J2SE-API unter Berücksichtigung der Sicherheitsregeln
- Auch komplexe Anwendungen möglich



Beispiel



The screenshot shows a Mozilla Firefox browser window titled "KI-I: Umrechnung von Maßeinheiten - Mozilla Firefox". The address bar shows the file path "file:///C:/Daten/iktforum". The page content includes a title "Umrechnung von Maßeinheiten", a description of the applet's purpose, a link to the calculator, and a form with input fields for "Wert in mm:", "Ziel-Maßeinheit" (with radio buttons for "Inch", "Pica", and "Punkt"), a "Berechnen" button, and an "Ergebnis:" field. A "Hilfe" link is at the bottom, and a status bar shows "Fertig".

KI-I: Umrechnung von Maßeinheiten - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

file:///C:/Daten/iktforum

KI-I: Umrechnung von Maßeinheiten

Umrechnung von Maßeinheiten

Das folgende Applet dient der Umrechnung von Maßeinheiten. Es können Maße von mm in inch, pica und punkt umgerechnet werden.

[Zum Rechner](#)

Wert in mm:

Ziel-Maßeinheit

Inch

Pica

Punkt

Ergebnis:

[Hilfe](#)

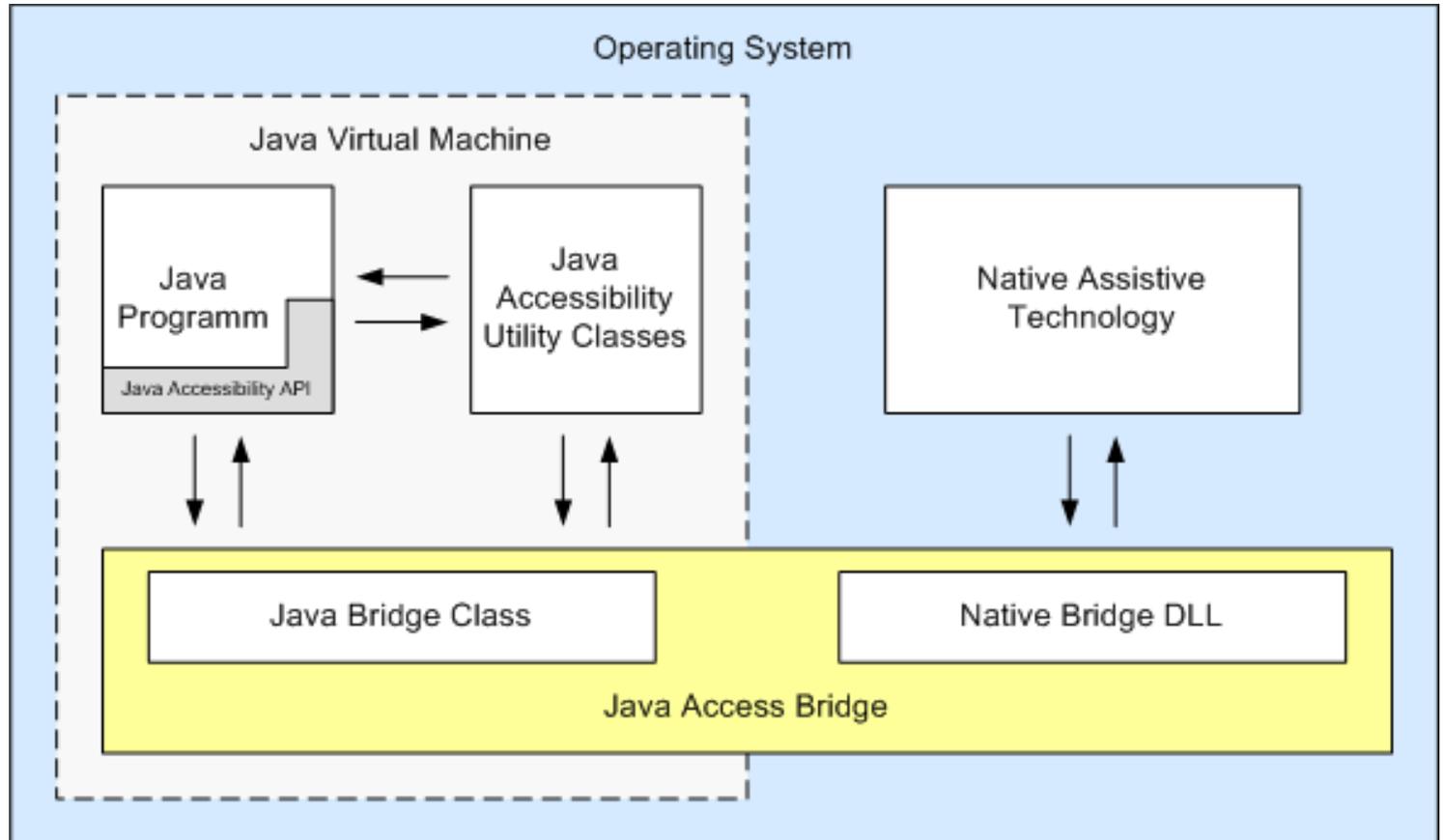
Fertig

Java Applets - Accessibility

- Interface `Accessible` aus *javax.accessibility*
 - Enthält nur die Methode „`getAccessibleContext()`“ welche den `AccessibleContext` der Komponente zurückliefert
 - `AccessibleContext` dient zum Setzen wichtiger Parameter:
 - Accessible Name (wird auch automatisch von statischem Text einer Komponente übernommen)
 - Accessible Role (meist passt die Standard-Role)
 - Accessible Description (wird auch automatisch vom ToolTip einer Komponente übernommen)
 - Etc.
- *javax.swing*: von fast allen Komponenten implementiert (!JSpinner)
 - Benötigt Java AccessBridge am Client
- *java.awt*: von einigen Komponenten implementiert
 - AccessBridge wird nicht benötigt



Access Bridge



FocusTraversal?

- Wechsel Webinhalt <-> Applet
 - Standardverhalten:
 - Focus automatisch im Applet: -> Webinhalt?
 - Focus automatisch im Webinhalt: -> Applet?
- Applet -> Webinhalt
 - Applet-Hack: Notausgang implementieren (Button mit `focusGainedEvent` -> Link nach Applet erhält Focus)
- Webinhalt -> Applet
 - HTML-Hack: Applet muss in den Taborder mit `TabIndex` „gezwungen“ werden;
`onfocus: document.appletID.requestFocus()`
 - Applet-Hack: *`requestFocus()`* in Methode *`init()`*

Wert in mm:

Ziel-Maßeinheit

Inch
 Pica
 Punkt

Ergebnis:

[Hilfe](#)



FocusTraversal

- FocusTraversalPolicy muss für umgekehrten FocusTraversal (Shift+Tab) implementiert werden
 - FocusTraversalPolicy muss für beide FocusTraversal-Richtungen festgelegt werden
 - `public java.awt.Component getComponentAfter(java.awt.Container focusCycleRoot, java.awt.Component cmp) {}`
 - `public java.awt.Component getComponentBefore(java.awt.Container focusCycleRoot, java.awt.Component cmp) {}`
- Zwei „Notausgänge“ im Applet erforderlich
 - Am Anfang: für Wechsel Applet->Webinhalt bei umgekehrtem FocusTraversal (auf fokusierbares Element vor dem Applet)
 - Am Ende: für Wechsel Applet->Webinhalt bei normalem FocusTraversal (auf fokusierbares Element nach dem Applet)



Applet-Einbettung in HTML

- `<object>`
 - Applet wird in allen Browsern aufgerufen (trotzdem Hack notwendig)
 - Kann valide gemacht werden! 😊
 - Kein FocusTraversal! ☹️
- `<applet>`
 - funktioniert recht gut in allen Browsern
 - FocusTraversal nur in IE & FF – mit folgender Einbettung: 😊

```
<applet code="Calculator.class" codebase="." id="Calculator" tabindex="2"
  onfocus="document.Calculator.requestFocus();">
  <param name="initial_focus" value="false" />
</applet>
```

HTML nicht valide;
 - `<applet>` ist deprecated! ☹️

Einschränkungen?

Kriterien	Erfüllbarkeit?
Tastaturbedienbarkeit	Gegeben (zumindest im IE & FF)
Ausgabe am Screenreader	Gegeben (zumindest im IE & FF)
Automatische Änderung der Schriftgröße	etwas kompliziert – muss ausprogrammiert werden
Automatische Änderung Vorder-/Hintergrundfarbe	etwas kompliziert – muss ausprogrammiert werden
...	...

Weitere Kriterien & Richtlinien in wenigen Minuten!



Fazit zu Applets

- Brauchbar in kontrollierter Umgebung
 - Z.B. Unternehmensweite Software
 - Mächtiges Werkzeug
 - Accessibility möglich 😊
- Barriere-Risiko in nichtkontrollierbarer Umgebung
 - Z.B. Internet
 - Probleme:
 - Viele unterschiedliche Java-Versionen
 - Unterschiedlichste Assistierende Technologien
 - Unterschiedliche Betriebssysteme
 - Unterschiedliche Browser
 - Etc.
 - Accessibility kann nicht garantiert werden! ☹️



Software Accessibility

- Drei wichtige Grundprinzipien:
 - Software sollte so entwickelt werden, dass sie für den breitest möglichen Benutzerkreis ohne weitere Hilfsmittel (Hard- und Software) benutzbar ist. (Direct Accessibility)
 - Software sollte so entwickelt werden, dass sie Accessibility Funktionen des Betriebssystems und mit assistierenden Technologien (Hard- und Software) zusammenarbeitet. (Dies ist speziell für Personen wichtig, die mit assistierenden Technologien arbeiten und Software nicht direkt benutzen können).
 - Die Dokumentation der Software, Training (Lernunterlagen) und Customer Support Systems sollten ebenfalls accessible sein.

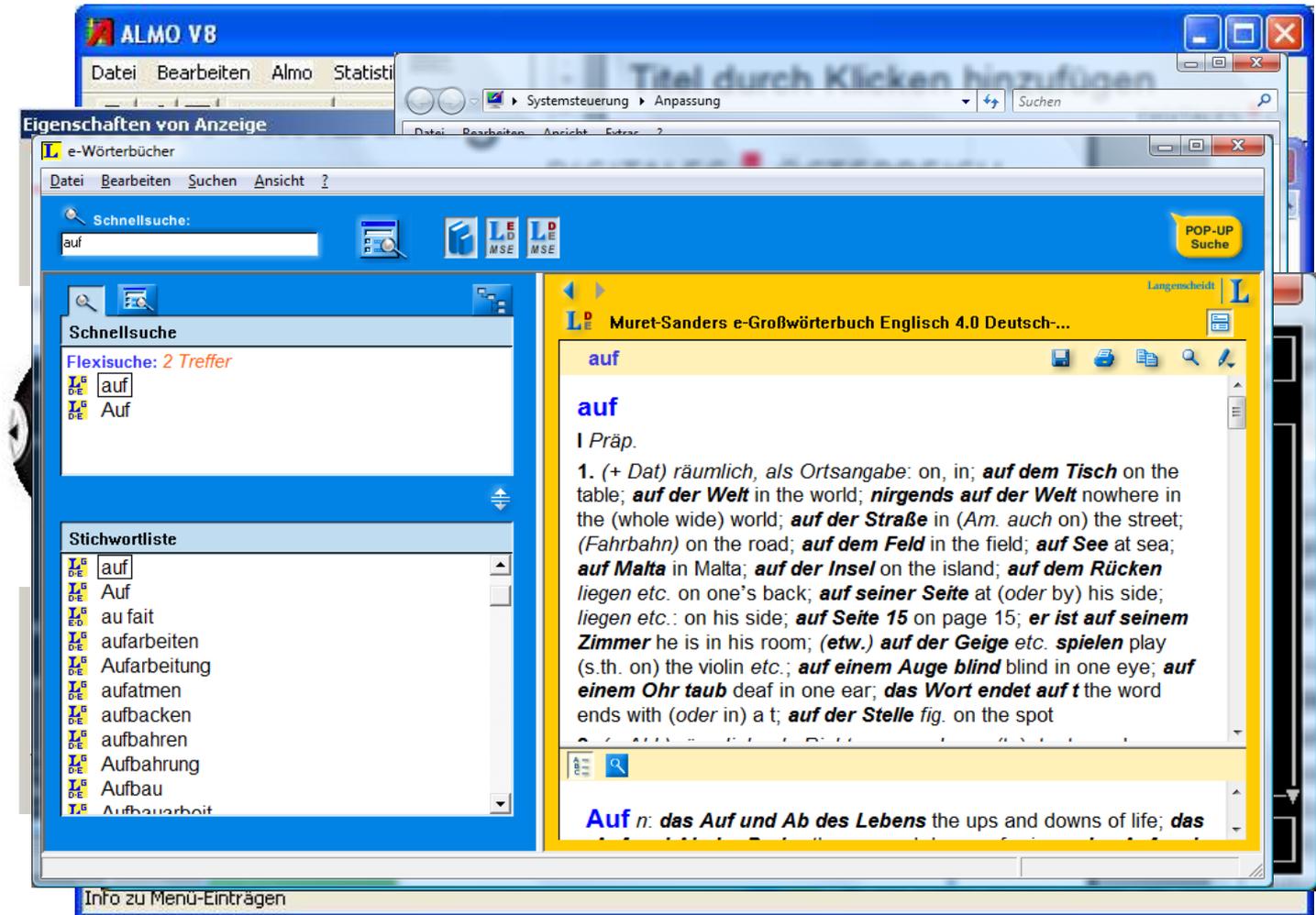


Software vs. Accessibility

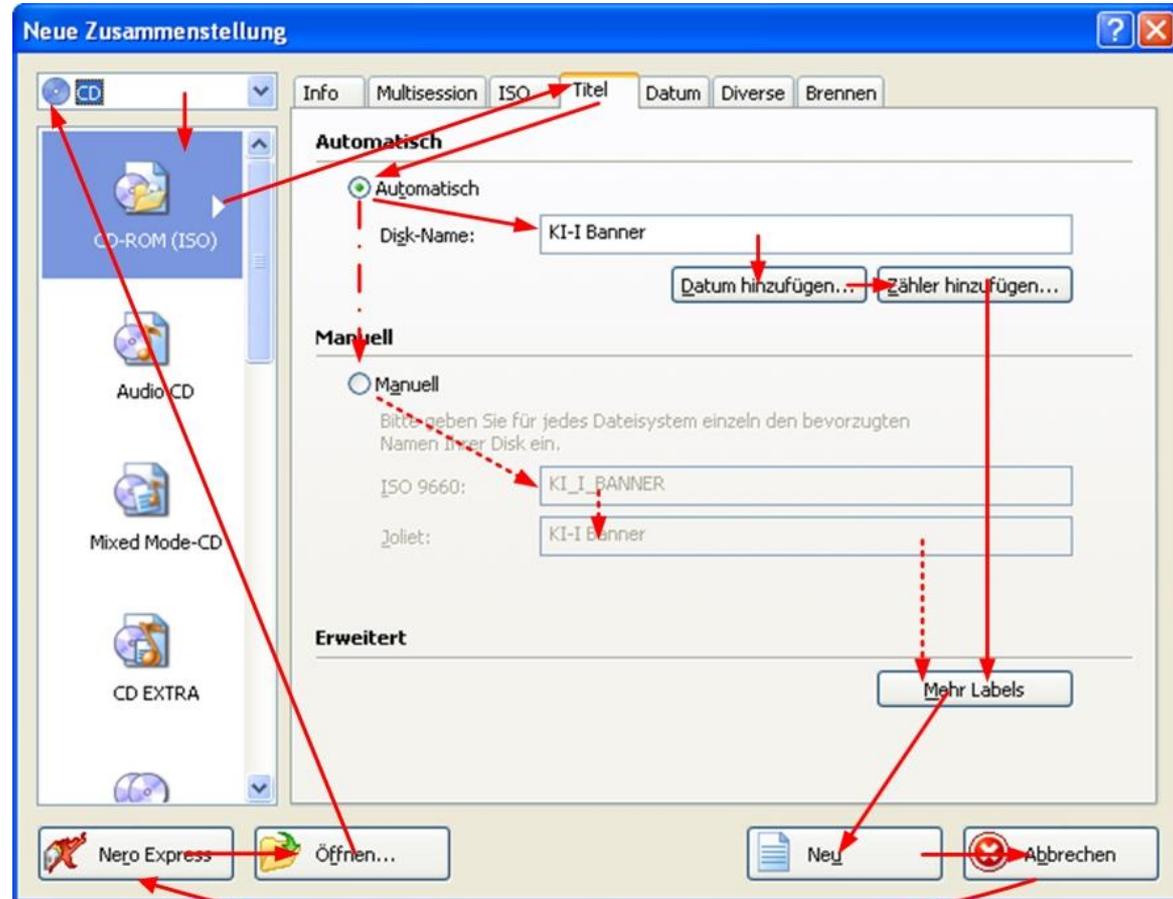
- Tragweite von Barrieren abhängig von der Art der Software
 - Systemsoftware: Gesamtes System (un)brauchbar
 - Anwendungssoftware: Einzelne Anwendung (un)brauchbar
- Benutzer sollte Software (in-)direkt an seine Bedürfnisse anpassen können
 - Farben/Kontraste
 - Schriftart/-Größe
 - ...
- Beachtung von Gestaltgesetzen, Styleguides, (Design Patterns)
- Strukturierung



Beispiele



FocusTraversal



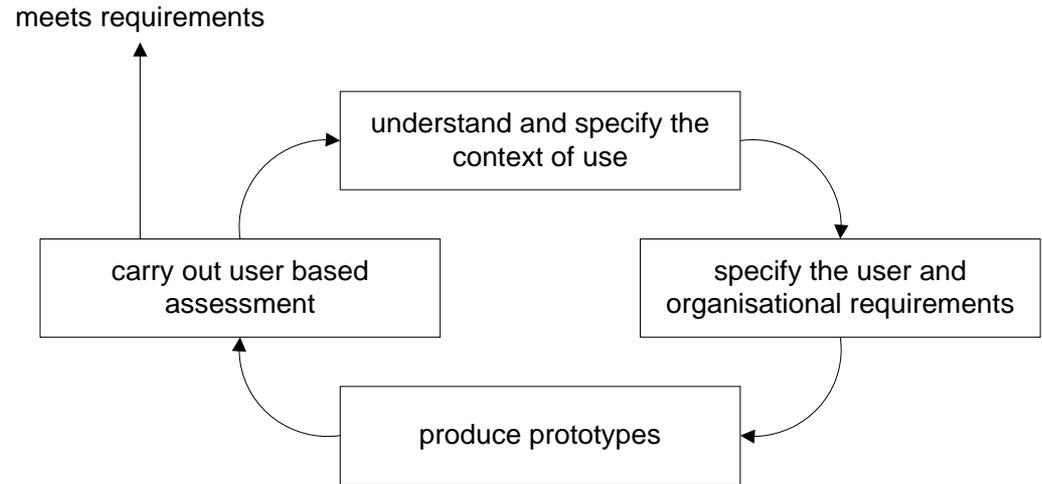
Richtlinien

- Usability von Software:
 - ISO 9241-x: Ergonomic requirements for office work with visual display terminals
 - ISO 9241-11: Ergonomic requirements for office work with visual display terminals - Guidance on usability
- Accessibility von Software:
 - ISO 9241-171: Ergonomics of human-system interaction - Guidance on accessibility for human-computer interfaces
 - W3C WAI Authoring Tool Accessibility Guidelines (ATAG)
 - W3C WAI User Agent Accessibility Guidelines (UAAG)
 - IBM software accessibility checklist (auch Java)
<http://www-03.ibm.com/able/guidelines/software/accesssoftware.html>



User Centred Design

- Ziel: Software, die genau dem entspricht, was Benutzer wollen/brauchen
- Iterativer Prozess:



- Wichtige Methode: Benutzertests
Schon der Test mit einer Person (mit Behinderung) kann gravierende Usability- und Accessibilitymängel aufdecken
- User Centred Design kann übrigens auch im Webdesign angewendet werden.

Übrigens ...

- Software-Accessibility ist wie auch Web-Accessibility im Behindertengleichstellungsgesetz verankert:
- **Behindertengleichstellungsgesetz §6 (5):** Barrierefrei sind bauliche und sonstige Anlagen, Verkehrsmittel, technische Gebrauchsgegenstände, **Systeme der Informationsverarbeitung** sowie andere gestaltete Lebensbereiche, wenn sie für Menschen mit Behinderungen in der allgemein üblichen Weise, ohne besondere Erschwernis und grundsätzlich ohne fremde Hilfe zugänglich und nutzbar sind.



Danke für Ihre Aufmerksamkeit
und Ihr Interesse.

gn@ki-i.at

www.ki-i.at

Java Accessibility

- Schon gesagt – Grundlegendes gleich wie bei Applets



Und was ist mit Microsoft?

- MSAA – Microsoft Active Accessibility
 - Ziele
 - Softwareprodukte, die MSAA unterstützen, sollen besser accessible sein
 - Assistierende Technologien, die MSAA unterstützen, sollen zuverlässiger und robuster sein
 - Über die MSAA bekommen Assistierende Technologien Informationen über Komponenten der GUI. Programme können auch Informationen zur Verfügung stellen
 - Typ des Objekts
 - Name des Objekts
 - Position des Objekts
 - Momentaner Zustand des Objektes
 - Zusätzlich bietet MSAA
 - Benachrichtigungen über Änderungen in der Benutzerschnittstelle via Windows Events
 - Navigation (räumlich und logisch)



Microsoft .Net Framework Accessibility

- Namespace Accessibility, mit Interface Iaccessible
- MSAA wird somit von .Net unterstützt
- Windows Forms dienen in .Net zur GUI-Programmierung
 - Alle in Windows Forms enthaltenen Steuerelemente implementieren IAccessible
 - Die Steuerelemente stellen so Informationen den Assistierenden Technologien zur Verfügung
- Bei Windows Forms Steuerelementen stehen folgende Eigenschaften zur Verfügung
 - AccessibleName
 - AccessibleDescription
 - AccessibleRole
- Weitere Eigenschaften von IAccessible, wie z.B. Location werden von den Windows Forms automatisch zugewiesen

