



## Intro

- Thomas Pfoch, Jg 1964, verh., sechs Kinder
- Informatiker (Dipl.)
- Seit 1992 Access
- Seit 2000 picoware gmbh
- *Wer gehört zu wem:* Commerzbank
- *Dt. Meldesystem für Infektionskrankheiten*  
Robert Koch-Institut (Gesundheitsberichterstattung  
Schweinegrippe, EHEC, HIV-Statistik)
- Weitere Infos: [www.picoware.de](http://www.picoware.de)
- PM: [thomas.pfoch@picoware.de](mailto:thomas.pfoch@picoware.de)



## Thema

Ideen für die Gestaltung von Access-Applikationen

**Frust:**

*[Kunde]* „Ach sieht man – das ist doch Access!“

**Motivation:**

- Herr Pfoch, funktional ist ihre Anwendung ja sehr schön...  
...aber können Sie sich nicht ein *etwas moderneres Design* einfallen lassen?
- Graue, starre Formulare mit dutzenden grauer Knöpfe
- Die Grenzen immer noch ein wenig zu erweitern



## Ergebnis: Komplexes Framework

- Tabellen- und Formular-Erzeugung aus Metadaten
- Datenbank-Time-Machine mit SQL-Server
- grafisch akzeptables UI-Framework
- projektbezogene Formulare nur noch einklinken
- Treeview-Navigation mit Steuerung über Skripte und SQL-Statements
- diverse grafische Spielereien zu Nutzer-Interaktion und Visualisierung
- haufenweise Tools und Gestaltungskonzepte
- Bibliotheks- und Projektfundus mit Feature-Paketen
- automatische Generierung von auslieferbaren Applikationen
- simples Setup- und Update-Konzept für Installationen in größeren Unternehmen



## Heute im Angebot:

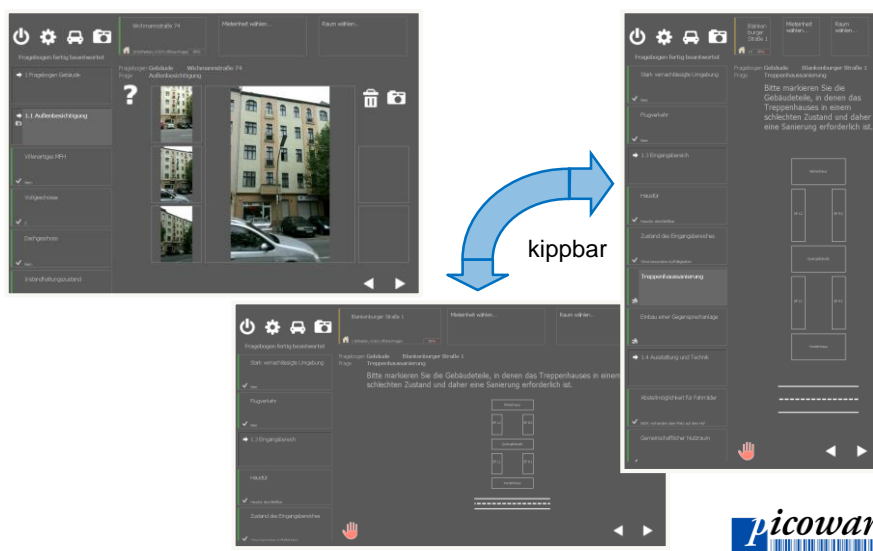
...nur ein wenig gefällige Grafik!

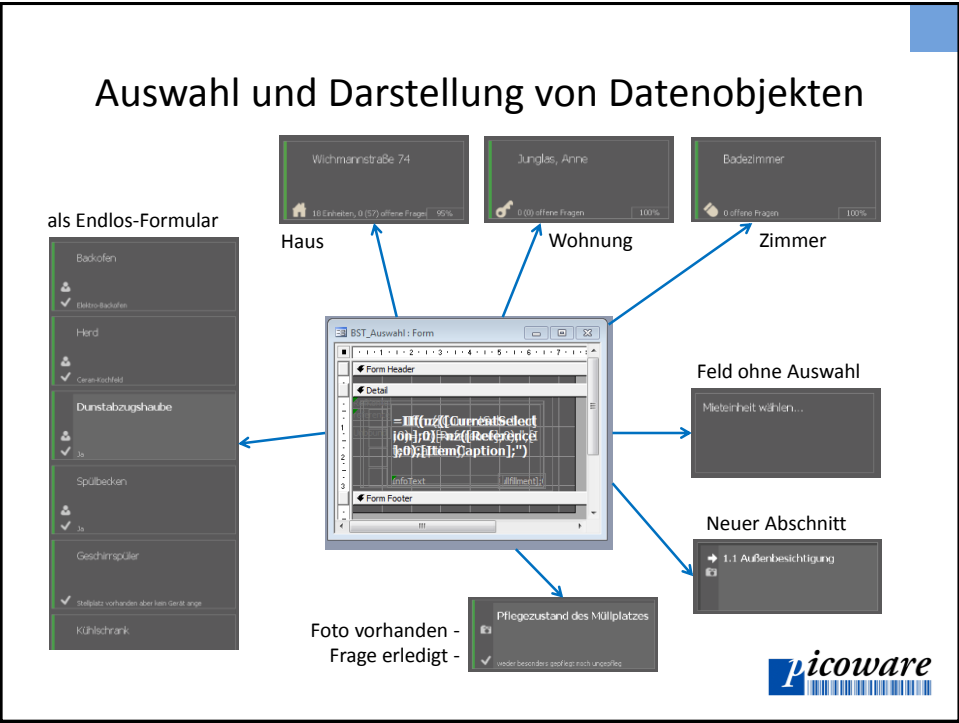
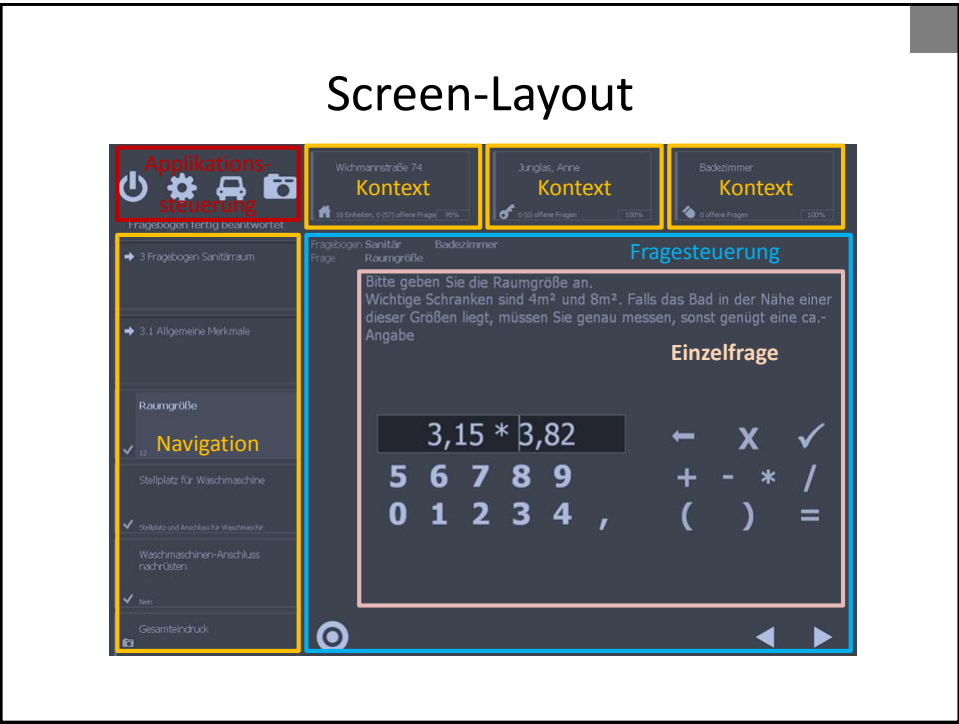
Erläuterung der Konzepte  
keine Details

...ein bisschen so wie „Europa in drei Tagen“



## Demo: Win8-App – fast & fluid...





## Aufruf Foto-App

```
Dim objShell As Object 'WshShell

Set objShell = CreateObject("WScript.Shell")
objShell.SendKeys "^{ESC}"
Sleep 100
objShell.SendKeys "Camera"      'bzw. "Kamera" bei dt.
Sleep 100
objShell.SendKeys "{ENTER}"
```

...das ganze natürlich innerhalb einer Fehlerbehandlung mit Prüfung, ob es wirklich Win8 ist, etc.



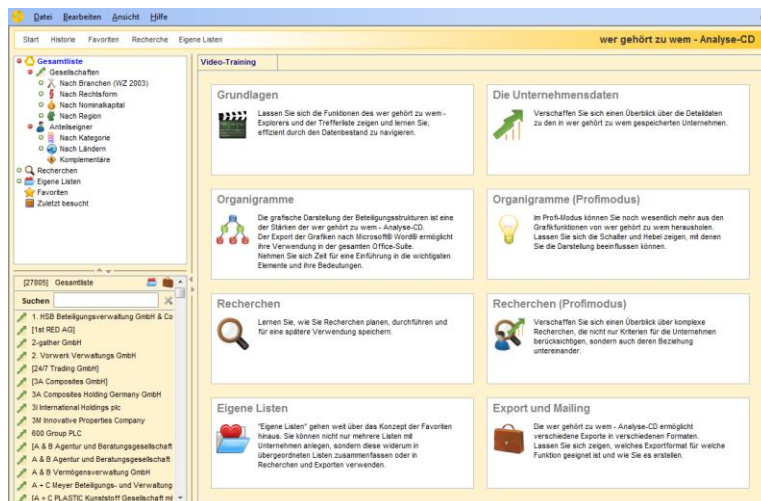
## Verwendete „Tricks“

- Bildschirmfüllendes, modales Popup ohne Rahmen
- Funktionsflächen (für OnClick) großzügig bemessen (Finger sind dick!)
- Scrollbars mittels Überlappung durch das rechtsseitige Element verstecken
- Alle Elemente ohne Rahmen verwenden
- Für Icons ein Image-Control verwenden (statt CommandButton) (OnClick!)
- Gewünschte Hintergrundfarbe für Icons bereits in der Bild-Quelle einstellen
- Auswahl-UFO (s.o.) hat einen flächendeckenden, transparenten CommandButton als oberstes Control
- Fragetyp-spezifische UFOs werden dynamisch eingeblendet und erfassen die Antworten. (Interface wegen Erweiterbarkeit nutzen)
- dynamische Größenanpassung in allen Forms

...und natürlich ein Haufen programminterner Kommunikation und Techniken (Interfaces, Events, Pattern)



## Teil II: wer gehört zu wem



## Demo: Dynamische Formulare

- Anpassung an Bildschirmgröße
- Teilbereiche verkleinern
- Teilbereiche aus- und einblenden
- Hierarchische Flächenstruktur
- Composite Pattern hält Controls und untergeordnete Teilflächen
- Splitter kennt linke und rechte (bzw. obere und untere) Fläche

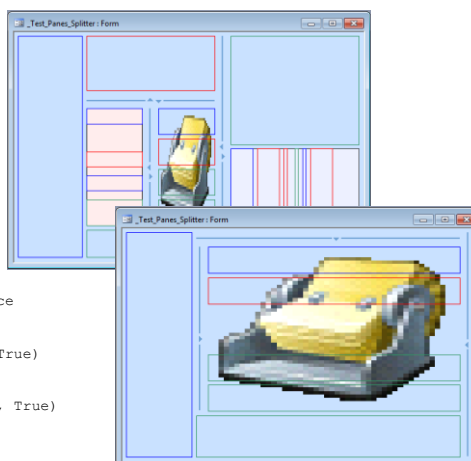
### Verwendung:

```
Set MyPane = New UI_Pane
MyPane.Init "Top", False, MySpace, MySpace
MyPane.AddControl Box1, 1, MySpace

Set MyPaneLeft = MyPane.AddPane("Left", True)
MyPaneLeft.AddControl Box2, 1, MySpace

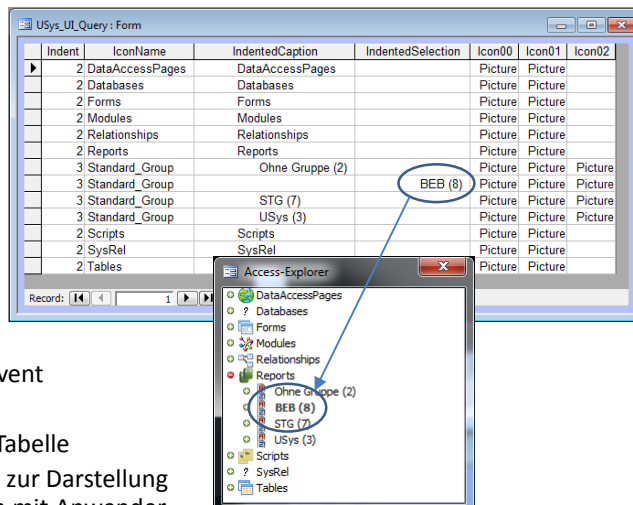
Set MyPaneRight = MyPane.AddPane("Right", True)
MyPaneRight.AddControl Box3, 1, MySpace

Set MySplitterH = MyPane.AddSplitter
Set MySplitterH.PaneLeft = MyPaneLeft
Set MySplitterH.PaneRight = MyPaneRight
```



## Demo: Treeview mit Bordmitteln

- Zeilen in Tabelle
- Text-Einrückung mit Leerzeichen
- Endlos-Form für Darstellung
- Icons als Picture über Left Join
- Mausposition ergibt Clickposition
- Clickposition ergibt Event
- Gesamter Baum in Tabelle
- Form ausschließlich zur Darstellung und Kommunikation mit Anwender



## Treeview - ein paar Tricks

- Endlosformular - Datenquelle enthält fertige Baumstruktur
- 16 Felder für Icons (BoundObjectFrame)
- Icons als Grafik über Left-Joins aus Grafik-Tabelle
- Einrückung des Knoten-Textes über Leerzeichen (3 pro Ebene)
- Unterscheidung normal / selektiert über zwei Spalten\* mit unterschiedlichen Formatierungen im EFO
- Einfärbung: vollflächiger BoundObjectFrame mit dynamisch erzeugtem Bildobjekt (16x16 Pixel) und Zoom
- Abdecken der Einfärbung vor den Icons mit weißen Kacheln

```
* iif(isSelected, "",Caption) as CaptionNormal,
  iif(isSelected, Caption, "") as CaptionSelected
```



## Demo: Skinning

Zentrale Steuerung des Layouts

- Alle offenen Formulare/Berichte passen sich automatisch an ein gegebenes Layout an
- Einheitliches Aussehen des UI
- Ggf. durch Benutzer anpassbar

Realisierung über eine

- Skinning-Klasse (wg. Events)
- Singleton-Pattern (zentrale Stelle)

Benötigt nur **sechs Zeilen Code** pro Form!

```
Private WithEvents MySkin
    As UI_Skinning-Token
```

```
Private Sub MySkin_SkinChanged()
    MySkin.SetLayout Me, False
End Sub
```

```
Private Sub Form_Load()
    Set MySkin = Skin
    MySkin_SkinChanged
End Sub
```



## Skinning Singleton

```
Public Property Get Skin() As UI_Skinning-Token
```

```
    Static SingletonSkin As UI_Skinning-Token
```

```
    If SingletonSkin Is Nothing Then
```

```
        Set SingletonSkin = New UI_Skinning-Token
```

```
    Endif
```

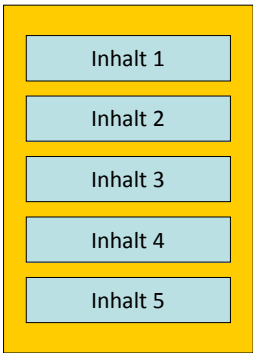
```
    Set Skin = SingletonSkin
```

```
End Property
```





# Demo: Auswahl-PopUps

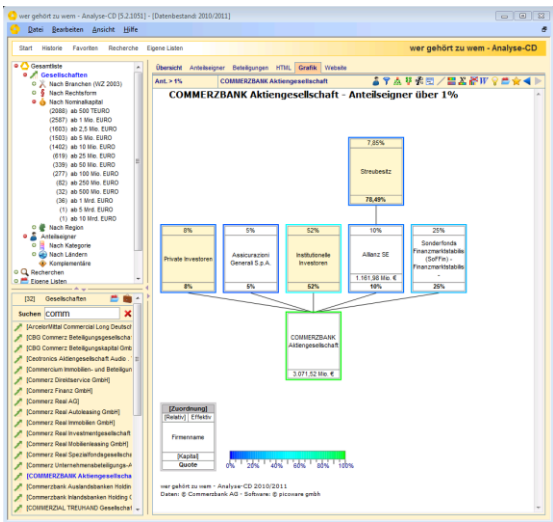


- modales PopUp-Form
- ähnlich einer ComboBox
- präsentiert eine Auswahlliste
- mögliche Datenquellen
  - Werteliste / Abfrage / anderes
- Ablauf:
  - Aufruf
  - Auswahlwerte holen
  - zur Anzeige aufbereiten
  - eigene Fläche anpassen
  - 2 sec auf Auswahl (Maus-Klick) warten
  - Auswahl oder NULL zurückliefern

...aber muss es immer Text sein?



# Demo: Suchen / Organigramme

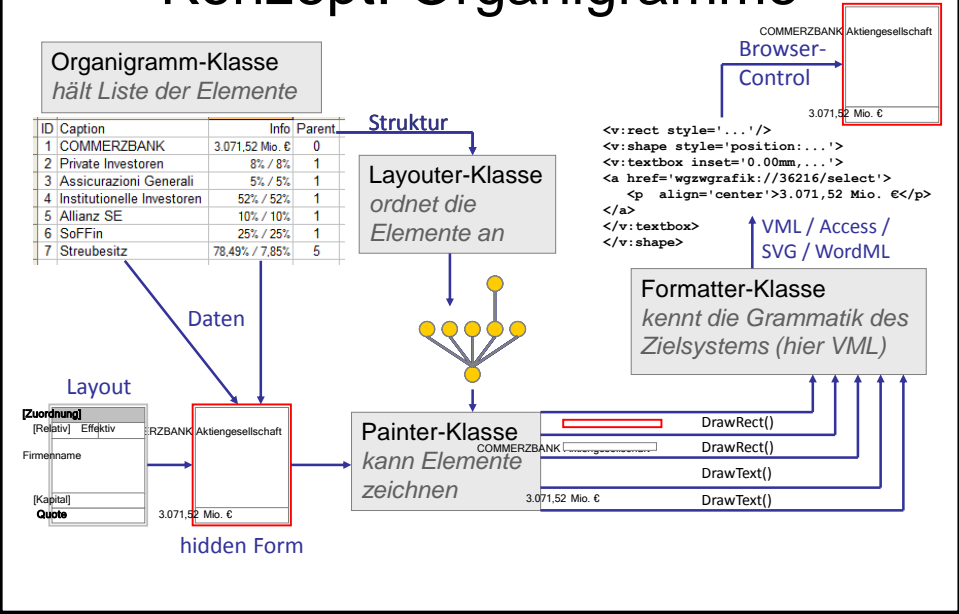


Die Realisierung verwendet fünf Komponenten:

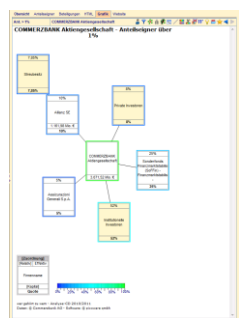
- Organigramm-Klasse
  - hält Liste der Elemente*
- Layouter-Klasse
  - ordnet die Elemente an*
- Painter-Klasse
  - kann Elemente zeichnen*
- Legend-Klasse
  - kann die Legende zeichnen*
- Formatter-Klasse
  - kennt die Grammatik der Zieldatei (hier VML)*



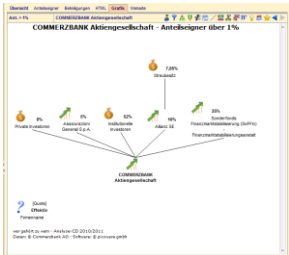
# Konzept: Organigramme



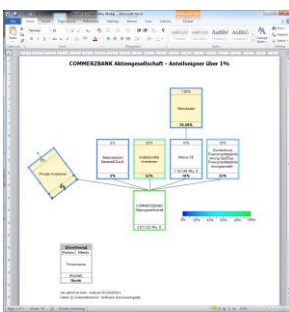
## Demo: Ändern von Komponenten



Ändern des Layouters

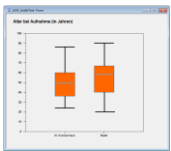


Ändern des Vorlage-UFOs



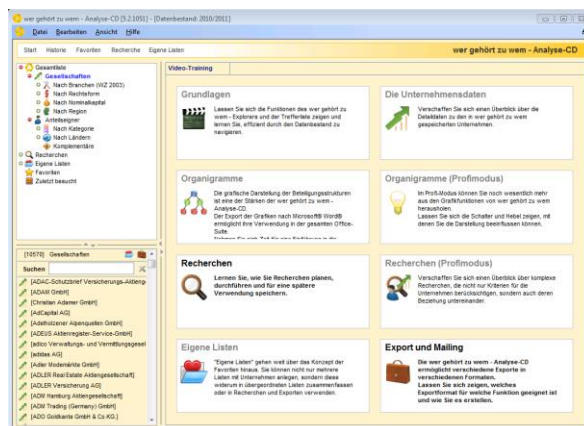
Ändern des Formatters

Add-On: Form 2 Word



## Demo: Video-Tutorial

- Handbücher liest keiner
- You-tube braucht einen Web-Zugang
- Lokale Videos benötigen viel Platz
- Bewegt-Bilder brauchen einen Player
- Gezieltes Springen zu einem gesuchten Inhalt ist in Videos schwierig
- Nebstehend acht Einstiegspunkte in ein visuelles Handbuch mit 35 MB für ca. 30 min



## Demo: Video-Tutorial

Bei Auswahl einer Szene werden Bild und Ton gestartet

Anzeige des Bildes:

```
If TutSet!Bild.FieldSize() <> 0 Then
    Field2File MyPath & FileNameBild, TutSet, "Bild"
    Me!Bild.Picture = MyPath & FileNameBild
End If
```

Ausgabe des MP3:

```
If TutSet!Ton.FieldSize() <> 0 And Nz(Me!ShowTon, 0) <> 0 Then
    MP3_Stop MyMp3Alias
    Field2File MyPath & FileNameTon, TutSet, "Ton"
    SetMP3Mode "Play", MyPath & FileNameTon
End If
```

Für MP3 gibt es eine Bibliothek, die u.a. Play, Stop, Pause und Resume über folgende API-Funktion realisiert

```
Private Declare Function mciSendString Lib "winmm.dll" _
    Alias "mciSendStringA" ( _
        ByVal lpzCommand As String, _
        ByVal lpzReturnString As String, _
        ByVal cchReturnLength As Long, _
        ByVal hwndCallback As Long) As Long
```

