

Formularmanagement

W. Riedel, TU Chemnitz, URZ

<mailto:w.riedel@hrz.tu-chemnitz.de>

 531 1422

Workshop, 14.-17.4.2003, Löbsal

Ausgangspunkt

- URZ-Formulare anpassen an TU-CD
- bessere Nutzbarkeit durch *alle* Nutzer
- vorhandene Defizite mit Acroread unter Linux
- Versuch eines „Brückenschlags“ zu PDF-Formularen der TU-Verwaltung

Erster Schritt: Layout

- CD enthält keinerlei Vorgaben für solche Dokumente
- deshalb Eigeninitiative: Benutzung des Brieflayouts
- Quellformat: $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Nutzerformat: PS bzw. PDF
- Ergebnisse: Loginkennzeichen, Softwareüberlassung

Bewertung:

- + Fortschritt gegenüber Vorzustand
- ? optisch nicht ideal, sinnvollen Alternative willkommen
- online nicht ausfüllbar
- ungeeignet für elektronische Unterschrift
- keine automatische Verarbeitung des Formulars durch URZ
- keine elektronische Archivierung der Daten durch Nutzer

Zweiter Schritt: HTML-Schnittstelle

- das einfachste plattformneutrale Format für Webanwendungen ist HTML
- Idee:
Nutzerschnittstelle (für Eingabe): HTML-Formular
Ergebnis: PDF (wie oben)
- Umsetzung: PHP-Skript

Formular Version 1

- Formularfelder entsprechend dem gewünschten Ergebnis
- gestalteter Text
- Submit-Button: Erzeugen eines druckbaren PDF-Files
(zum Unterschreiben und Verschicken per Post)

Version 1

Submit bewirkt:

- Übergabe der Formulardaten an ein (anderes) PHP-Skript
- Einfügen in ein \LaTeX -Template
- Aufruf von `pdflatex`
- Antwort des Webservers ist erzeugtes PDF-File

Ergebnis Version 1

Bewertung:

- + online ausfüllbar
- keine elektronische Archivierung der Daten durch Nutzer

Deshalb: Erweiterung der Version 1 \Rightarrow **Version 2**

- Speichermöglichkeit der Formulardaten \Rightarrow als XML (XFDF)
- Möglichkeit zum Laden vorhandener Formulardaten

Webschnittstelle

Ergebnis

Formulardaten

FDF/XFDF = Forms Data Format

- Entwicklung von Adobe
- Spezifikation ist veröffentlicht
- wird im Acrobat benutzt (zum Speichern von Formulardaten)
- FDF: ASCII, kryptisch
XFDF: XML-like

Bewertung:

- + alle Ziele erreicht (außer Signatur) 😊
- ? es bleibt noch „etwas“ Fleißarbeit, für alle URZ-Formulare entsprechende PHP-Skripte und \LaTeX -Templates zu schreiben

Aber:

- keine Lösung für die vorhandenen Formulare der Verwaltung machen 😞

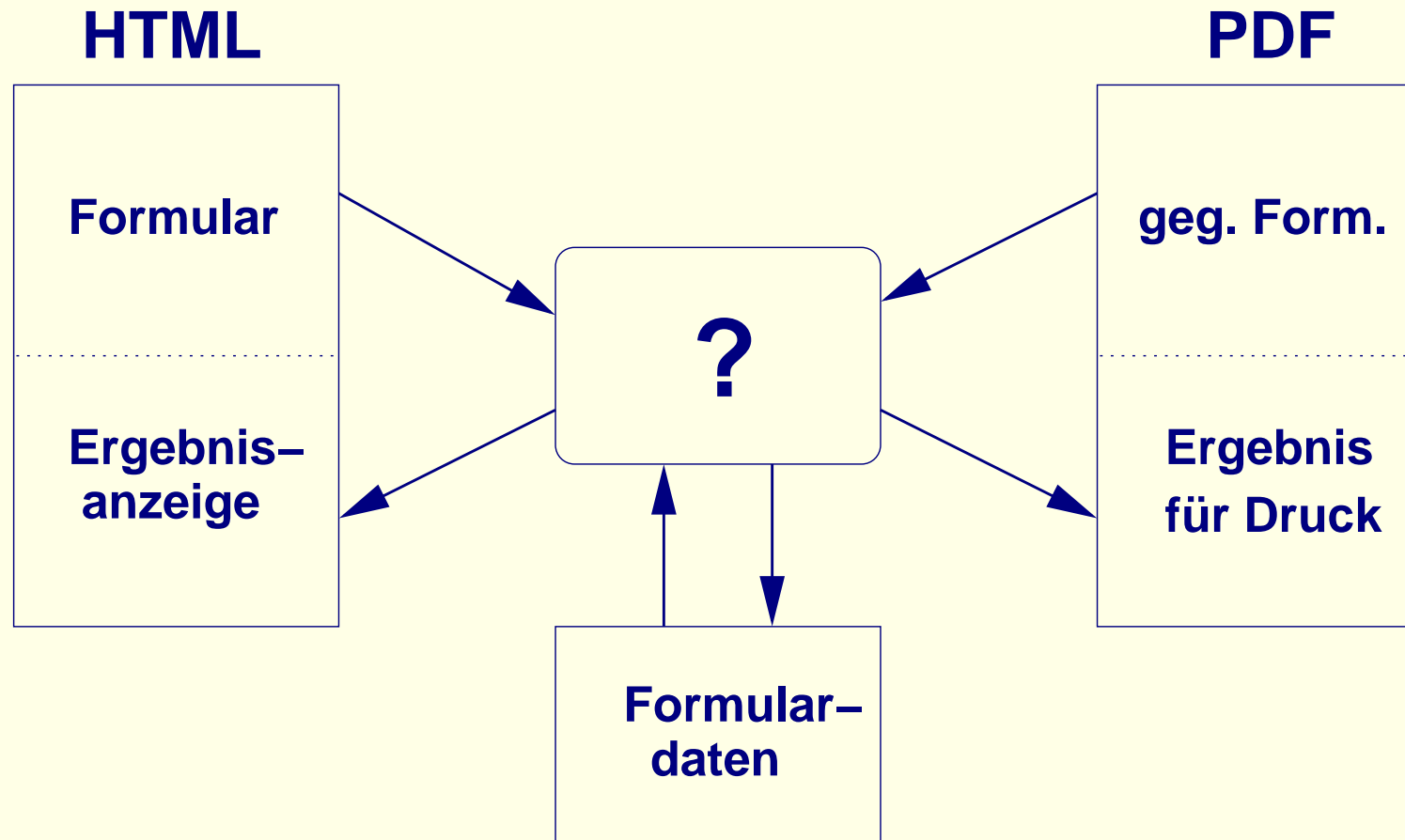
Dritter Schritt:

Vorhandene PDFs nutzbar machen

Schlüssel zum Erfolg: PDFlib

- einzige Software (außer Acrobat), die PDF modifizieren kann
- Entwicklung von Thomas Merz
- (leider) keine freie Software, sondern kommerzielle Lizenz
- größtenteils kostenfrei nutzbar
- Schnittstellen für C, C++, Perl, Tcl, Python, PHP, ...

Idee:



1. Realisierungsversuch:

HTML-Formular:

- Tabelle, mit PDF-Formular als Bildhintergrund
- „an den richtigen Stellen“ HTML-Eingabefelder, Checkboxen etc.

Quelltext → Anzeige

Verarbeitung:

- PHP-Funktionen aus PDF-Modul (PDFlib)

Quelltext-Gerüst → Ergebnis

Bewertung:

- Tabellenkonstruktion sehr aufwändig
- Positionierung ungenau
- Positionierung im PHP sehr aufwändig

2. Realisierungsversuch für HTML-Seite:

- absolute Positionierung von Elementen mittels `div`-Konstruktionen:

```
<div style="position:absolute; ...>
```

Quelltext → Anzeige

Bewertung:

- Bestimmung der richtigen Positionen sehr aufwändig
- Positionierung im PHP gleich aufwändig

3. Realisierungsversuch für HTML-Seite:

- Benutzung des Hintergrundbildes als verweis-sensitive Grafik (map)

Quelltext → Anzeige1 → Anzeige2

Bewertung:

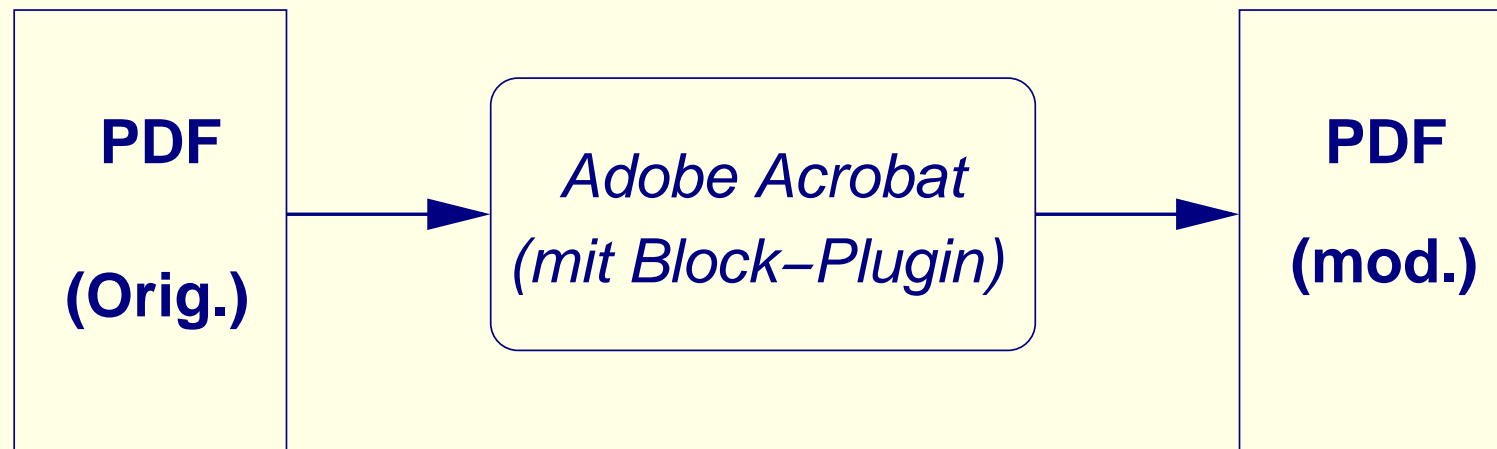
- Bestimmung der richtigen Positionen sehr aufwändig
- Bedienung aufwändig (jedes Feld einzeln eingeben und verarbeiten)
- Positionierung im PHP gleich aufwändig

Vierter Schritt: PDF-Blocks

- PDFlib Version 5: seit 3 Wochen freigegeben
- Konzept der „Blocks“ (allgemeiner als Formularelemente)
- ein Block entspricht einem Formularelement
- die Blöcke werden über Namen identifiziert (keine Positionsangaben nötig)
- **werden automatisch bestimmt:** Acrobat-Plugin

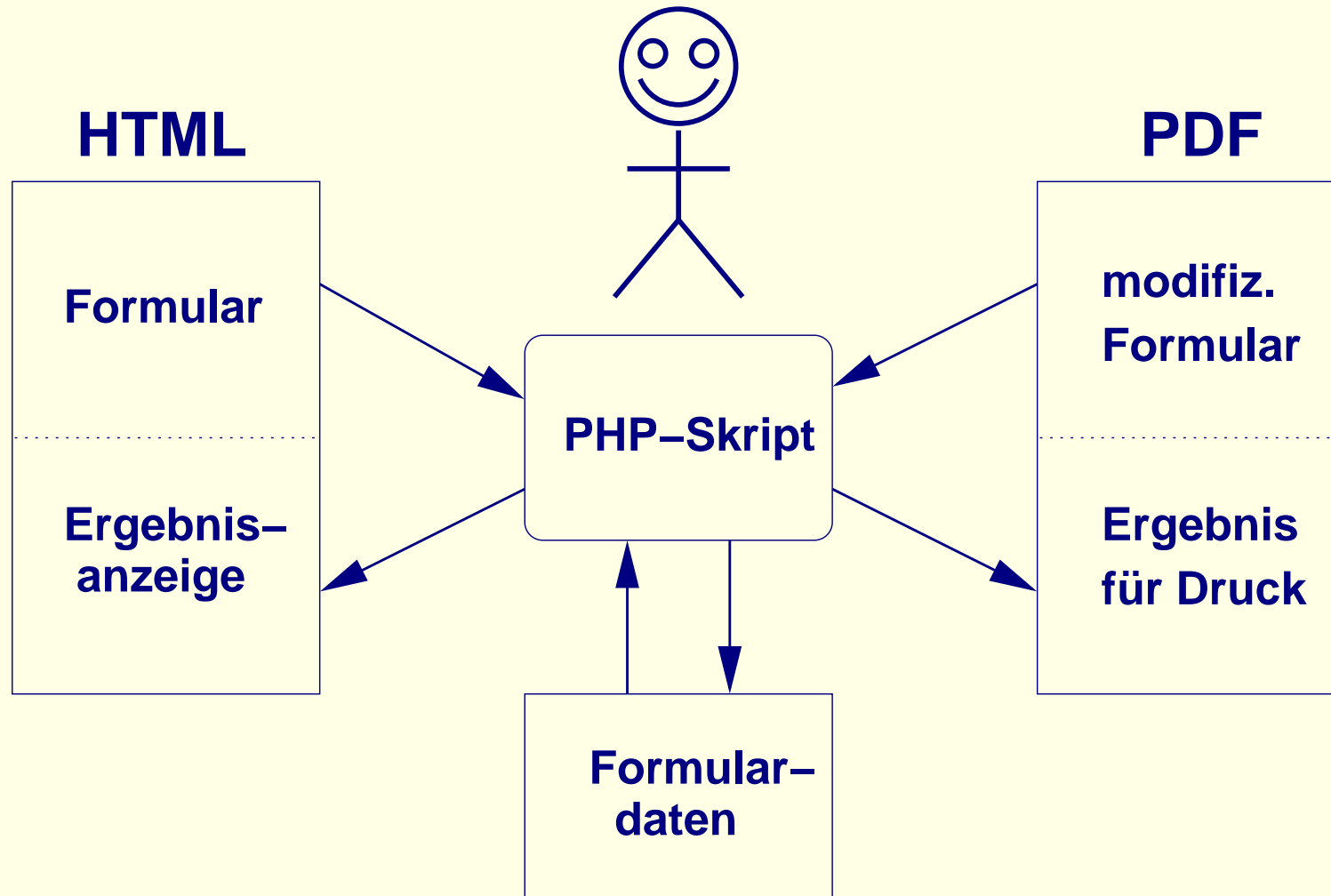
Damit ergibt sich folgender Verarbeitungsweg:

1. Administrator



- einmalig
- Anzeige der Formularfelderbezeichnungen

2. Nutzer (beliebig oft)



PHP-Skript:

- verarbeitet Eingaben der (HTML-)Formularfelder
- speichert diese als XFDF
- zuordnet die Werte den (PDF-)Formularfeldern zu
- Ausgabe des PDF-Files

Quelltext → Ergebnis

Webformular Dienstreiseantrag → Komplettes Ergebnis

Bewertung:

- + alle (realistischen) Ziele erreicht
- ? Kosten
- Programmieraufwand zur Entwicklung der HTML-Formulare