



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich



Institut für  
Technische Informatik und  
Kommunikationsnetze

**Matthias Keller**

# **Web Based Services (WBS)**

Semesterarbeit SA-2006-09

Wintersemester 2005/2006

Tutor: Marcel Baur

Supervisor: Prof. Dr. Bernhard Plattner

# Abstract

Das print on demand System „Web Based Services“ (WBS) des AMIV-Verlages ermöglicht es den Studierenden der ETH Zürich, Sammlungen früherer Prüfungen sowie Skripte über das WWW zu bestellen. Das derzeitige WBS-System ist veraltet und schwer zu warten. Ziel dieser Semesterarbeit ist es, die Grundlagen für ein Redesign zu erarbeiten. Anhand des Ist-Zustandes werden die Anforderungen der einzelnen Interessenvertreter dokumentiert und priorisiert. Auf der Basis dieses Anforderungskatalogs wird ein Konzept für die neue Applikation erstellt. Einige spezifische Teile des Konzepts werden exemplarisch implementiert, um dessen Tauglichkeit zu demonstrieren. Das erarbeitete Konzept wird darauf untersucht, ob es die erarbeiteten Anforderungen erfüllt. Schliesslich wird ein Ausblick auf offene Punkte und die nächsten Schritte gegeben.

The print on demand system "Web Based Services" (WBS) by 'AMIV-Verlag' allows students from ETH Zurich to order collections of former exams and scripts over the WWW. The current System is out of date and hard to maintain. The target of this semester thesis is to develop the base for a redesign. Given the status quo, the requirements of all involved parties are documented and prioritized. Based on these requirements, a concept of the new application is developed. Some specific parts of this concept are exemplary implemented to demonstrate their suitability. The concept is analyzed if it meets the requirements. Finally an outlook is given over the open tasks and the next steps.

# Inhaltsverzeichnis:

<b>1</b>	<b>AUFGABENSTELLUNG</b>	<b>4</b>
<b>2</b>	<b>ANALYSE</b>	<b>6</b>
2.1	AKTUELLES SYSTEM	6
2.2	BETEILIGTE PARTEIEN	8
2.3	PROBLEME	8
2.4	ANFORDERUNGEN	9
<b>3</b>	<b>KONZEPT</b>	<b>13</b>
3.1	ÜBERSICHT	13
3.1.1	<i>Dokumentenmanagement</i>	13
3.1.2	<i>Bestell-Interface</i>	18
3.1.3	<i>Fachvereins-Interface</i>	20
3.1.4	<i>Administrations-Interface</i>	22
3.2	MODULE	24
3.2.1	<i>Login</i>	24
3.2.2	<i>Druckablauf</i>	27
3.2.3	<i>Reservationssystem</i>	33
3.2.4	<i>Printdaemon</i>	34
3.2.5	<i>Fachmanager</i>	37
3.2.6	<i>FIBU Modul</i>	39
<b>4</b>	<b>IMPLEMENTIERUNG</b>	<b>42</b>
4.1	PRINTERSTEUERUNG	42
4.2	ESR (E-BANKING)	44
<b>5</b>	<b>EVALUATION UND AUSBLICK</b>	<b>46</b>
<b>6</b>	<b>SCHLUSSFOLGERUNGEN</b>	<b>47</b>
<b>A</b>	<b>ALTES DATENBANK-SCHEMA</b>	<b>48</b>
<b>B</b>	<b>NEUES DATENBANK SCHEMA</b>	<b>55</b>
<b>C</b>	<b>REFERENZEN</b>	<b>56</b>

# 1 Aufgabenstellung

Web Based Services (WBS)<sup>1</sup> bietet den StudentInnen der ETH Zürich (und künftig auch der anderen Hochschulen) einen einfachen und komfortablen Weg, Sammlungen von vorjährigen Basis-, Bachelor und Masterprüfungen sowie Skripte von Vorlesungen zu bestellen. Bestellte Dokumente werden bei WBS ausgedruckt und können dort abgeholt werden. WBS wurde im Jahr 2000 vom AMIV-Verlag (ehem. Kommission des Akademischen Maschinen- und Elektro-Ingenieur-Vereins AMIV) in Betrieb genommen. In den 5 Betriebsjahren wurde die Druckkapazität sowie die Menge an druckbaren Dokumenten (1800 Dokumente, Stand So05) dank einer stetig wachsenden Nachfrage kontinuierlich gesteigert. Ursprünglich nur für das Departement D-ITET konzipiert, enthält WBS nun Dokumente für 14 Departemente. Die Dienstleistung wurde über die Jahre mit Zusatzservices wie Authentifizierung via nethz<sup>2</sup> sowie der Möglichkeit, sich die Bestellung mit Rechnung zuschicken zu lassen erweitert.

Seit Inbetriebnahme des WBS wurden bereits über 1.5 Mio. Seiten gedruckt und die Tendenz ist steigend. Bereits ist die Dienstleistung nahtlos in den Studentenalltag integriert. Ein Studentenleben ohne WBS ist beinahe undenkbar geworden und der Anforderungsdruck seitens der Studenten für ein perfekt funktionierendes und angepasstes System steigt.

Doch mit der wachsenden Popularität kam auch der Wunsch nach neuen Funktionen und Umstrukturierungen der Studiengänge aufgrund der Bologna-Reform machen ein Redesign des Systems notwendig. Nach Abschluss der Neulancierung wird das System der VSETH-Kommission Student Printing on Demand (SPOD)<sup>3</sup> übergeben, welche dann den laufenden Betrieb übernimmt.

Ziel dieser Arbeit ist das angesprochene Neudesign von WBS.

## Ablauf

Diese Semesterarbeit ist in vier Phasen unterteilt:

1. Da verschiedene Interessenvertreter involviert sind, soll in einer ersten Phase **Analyse** zunächst der Ist-Zustand und die Anforderungen der verschiedenen Interessenvertreter aufgenommen werden. Die Anforderungen sind zu analysieren, in Absprache mit den Beteiligten zu priorisieren und zu dokumentieren.
2. Anhand der dokumentierten Anforderungen soll in einer zweiten Phase ein **Konzept** für die Technik und die Prozesse erstellt werden.
3. Die Tauglichkeit des technischen Konzepts soll dann exemplarisch in der dritten Phase **Implementierung** demonstriert werden.
4. Schliesslich sollen die Ergebnisse in einer vierten und letzten Phase dokumentiert werden.

## **Deliverables**

1. Schriftlicher Report welcher das laufende System beschreibt, die analysierten Anforderungen priorisiert, das technische Konzept und die zugrundeliegenden Prozesse aufführt und die Umsetzung dokumentiert.
2. Prototypische Implementierung. Diese soll demonstrieren, dass die erarbeiteten Konzepte umsetzbar sind und funktionieren.
3. Nach Abschluss jeder Projektphase ist eine kurze Präsentation zu halten
4. Nach Abschluss der ganzen Semesterarbeit ist eine Schlusspräsentation zu halten

## **Betreuung**

Marcel Baur betreut diese Semesterarbeit als Assistent. Ein Vertreter von SPOD wird die Semesterarbeit zusätzlich begleiten.

## 2 Analyse

### 2.1 Aktuelles System

Das aktuelle System ist unübersichtlich und schwer zu warten. Es besteht aus diversen PERL-Scripts und Marcel Strittmatter, der das WBS ursprünglich programmierte und seither auch alleine wartete und erweiterte, ist der einzige der das System versteht.

Es gibt keinerlei schriftliche Dokumentationen über Abläufe oder Einstiegspunkte.

Die einzelnen Dokumente sind im lokalen Dateisystem des Servers abgelegt unter dem Namen der Vorlesung; gruppiert in Verzeichnissen für jedes Departement.

Z.B.: /www/wbs/dokumente/d-itet/vordips/2005he/vd2

Das System erfüllt zwar in Grundsätzen die wichtigsten Anforderungen von damals; ist aber vor allem zu wenig flexibel anpassbar auf die Änderungen die sich mit anderen Departementen, den neuen Master-Studiengängen etc ergeben haben da es ursprünglich nur für das D-ITET Departement geplant war.

Dem System zugrunde liegt ein Linux-Server, der im CDR-Büro steht, der für das Drucken zuständig ist sowie den Webserver beherbergt. Bedient wird das System über einen Mac mit einem Barcode Scanner. Mit diesem werden dann die Legis von Kunden abgescannt.

Wenn Bestellungen fertig ausgedruckt sind werden sie vom System einem von etwa 120 Ablagefächern zugewiesen wo sie dann zur Abholung bereit liegen.

#### **Dateien**

Die derzeitige Applikation besteht aus folgenden Dateien auf dem Server:

/www/wbs/apps

|-- bin

| |-- backup: script zur Erzeugung von Backups der Datenbank (per cron)

| |-- cleartemps: löscht temporäre Einträge in der Datenbank (per cron)

| |-- code39tops: erzeugt einen code39 Barcode in Postscript

| |-- count\_pdf\_pages: zählt die Seiten eines PDF-Dokumentes

| |-- erinnerungsmail: erinnert Kunden über nicht abgeholte Bestellungen

| |-- generate\_pdfs: erzeugt von jedem ps file ein PDF

| |-- gnuccash\_export.pl: exportiert Einträge der Kasse für gnuccash<sup>4</sup>

| |-- import\_dokumente.pl: importiert Dokumente aus einem tgz

| |-- legicheck: überprüft eine Legi-Nummer auf ihre Gültigkeit

| |-- mahnung1\_wbs.pl: erzeugt Mahnungen

- | |-- mahnung1\_wbs\_mail.pl: erzeugt Mahnungen welche per E-Mail versandt werden
- | |-- mail\_newdocs: informiert Kunden per Mail über neue Dokumente
- | |-- parse\_esr\_file.pl: liest ein ESR-File und gibt lesbare Informationen aus
- | |-- print\_files.pl: Daemon der die Bestellungen ausdruckt
- | |-- rechnung\_wbs.pl: erzeugt Rechnungen
- | |-- tiff2wbs: erzeugt Postscript Datei aus einem Ordner mit TIFFs
- | |-- update\_files.pl: synchronisation db <-> document tree
- | `-- wbsmv: verschiebt Dokumente mit anpassen der Datenbank
- |-- lib
- | |-- authentication.pl: Funktionen für die Authentisierung (n.ethz, WBS)
- | |-- date\_lib.pl: Datumsmanipulations-Routinen
- | |-- html\_lib.pl: erzeugt HTML Code
- | |-- legi\_lib.pl: überprüft die Legi-Nummer auf Gültigkeit
- | |-- log\_lib.pl: Logfile-Routinen
- | |-- mail-lib.pl: E-Mail-Routinen
- | |-- md5\_lib.pl: berechnet MD5-Hashes
- | |-- ocrb726.ps: Schriftart für Einzahlungsscheine (OCR-B)
- | |-- overlib\_mini.js
- | |-- print\_lib.pl: Druckfunktionen
- | |-- ps\_header.ps: WBS spezifischer Postscript-Header
- | |-- sql\_base.pl: SQL Basis Funktionen
- | |-- sql\_lib.pl: SQL Funktionen
- | |-- utils\_lib.pl: Funktionen zum runden
- | `-- wbslogo.eps: WBS logo
- /www/wbs/admin.wbs.ethz.ch
- |-- admin.cgi: Script zur Administrierung von WBS
- |-- css: CSS-Dateien
- |-- esr: zum betrachten von ESR Dateien
- |-- fileadmin: file browser
- |-- img: Bilder
- |-- index.cgi -> admin.cgi
- |-- kasse: Manipulation der Kasse
- |-- mahnungen.cgi: erzeugen von Mahnungen -> PDF
- |-- newfiles: verwalten neuer Dokumente
- |-- print.cgi: ausdrucken des Dokumentbestandes

```
`-- templates: HTML Templates
/www/wbs/wbs.ethz.ch
|-- css: CSS Dateien
|-- img: Bilder
|-- index.html: Startseite
|-- profil.rdf
|-- robots.txt
|-- shop
| |-- banners: Werbebanner
| |-- design.css: generelle CSS Datei
| |-- design_unix.css: CSS Datei für UNIX-Browser
| |-- img: Bilder
| |-- index.cgi: WBS Hauptsript
| |-- libs: Overlib Scripts
```

## **2.2. Beteiligte Parteien**

Mit dem System interagieren derzeit folgende Parteien:

- Student
- Fachverein / VSETH
- SPOD

Die Institute der ETH Zürich sind nicht direkt beteiligt. Da die zuständigen Assistenten häufig wechseln und die Nachfolger wieder erneut eingeschult werden müssten, würde dies ein zu grosser Personalaufwand bedeuten. Dennoch können die Institute über SPOD-Mitarbeiter Vorlesungsskripte einfügen.

## **2.3 Probleme**

Das System WBS wurde ursprünglich nur für den Studiengang Elektrotechnik (heute: Informationstechnologie und Elektrotechnik) entwickelt. Durch die fehlende Unterstützung für verschiedene Departemente, Studiengänge und Prüfungsarten genügt es den heutigen Ansprüchen nicht mehr. Eine solche Unterstützung war nie vorgesehen.

Vor allem die Umstellung zu Bachelor/Master Studiengängen aufgrund der Bologna-Reform verursachen Probleme, da das System nur für Vordiplome und Schlussdiplomprüfungen ausgelegt war. Zudem sind die Prüfungsmodi nicht in allen Studienrichtungen identisch. Im aktuellen System können diese studiengangspezifischen Eigenschaften nicht korrekt abgebildet werden.

Auch existiert keine zentrale Konfiguration für globale Einstellungen wie z.B. der Preis pro Seite usw. Die meisten Werte sind an diversen Orten im Code fest einprogrammiert und können nur mit erheblichem Aufwand angepasst werden.

Probleme gibt es auch beim Fachmanager, der die fertig ausgedruckten Bestellungen einem Ablagefach zuweist bis sie abgeholt werden. Gerade vor Prüfungen gibt es sehr viele Bestellungen und die Fächer quellen über. Oftmals sind dann mehrere Fächer von einem einzigen Besteller belegt – das ist ziemlich umständlich. Es gibt ausserdem gelegentliche Inkonsistenzen wenn eine Bestellung gelöscht oder geändert wird. Dies resultiert darin, dass gewisse Fächer als belegt markiert bleiben obwohl es die Bestellung nicht mehr gibt.

Bei Fehlern (verursacht durch das System, den Kunden oder den Betreiber) im Bestellungsablauf ist es praktisch unmöglich, über das Verkaufsinterface in den Ablauf einzugreifen und Fehler zu beheben. Es existieren keine Möglichkeiten, den Ablauf einer Bestellung (dies beinhaltet die Bestellung selber und damit verbundene Abläufe wie das Rechnungswesen) gegebenenfalls zu ändern oder sie zu stornieren.

Die Möglichkeit der Rechnungsstellung wurde erst nachträglich ins System eingebaut und somit fehlen Mechanismen zur Kontrolle von Zahlungseingängen, zur automatischen Erstellung von Mahnungen und dem Export von Konto- und Kassendaten für die Finanzbuchhaltung GnuCash.

Ausserdem ist die Applikation sehr schlecht wartbar, da nur Marcel Strittmatter das System im Detail kennt und bei Bedarf in die Datenbank oder den Programmablauf eingreifen kann. Ist er abwesend, kann wenig unternommen werden, um ein Problem zu lösen.

Das bisherige Datenbank-Schema findet sich in Anhang A

## **2.4 Anforderungen**

Da die Applikation von mehreren Personen programmiert wird soll sie modular aufgebaut und auch während der Programmierung sauber dokumentiert werden.

Damit sollen auch mehrere Personen mit dem Code sowie der Systemadministration vertraut sein, um bei Problemen eingreifen zu können.

Das System soll resistent sein gegen Forward/Back-Button Probleme, z.B. durch die Verwendung eines speziellen Frameworks wie PHRAME<sup>5</sup>

Nachfolgend aufgeführte Anforderungen die nicht weiter klassifiziert sind, sind wichtig und *müssen* implementiert werden.

Anforderungen die mit 'nice to have' markiert sind sind keine zwingenden Anforderungen ans erste Release. Diese könnten jedoch dann in einem weiteren Schritt eingeführt werden.

Gestrichene Punkte bezeichnen Anforderungen, die bei der Evaluation als unnötig oder zu aufwändig klassifiziert und daher gestrichen wurden.

## **Funktionsumfang für den Kunden (Student)**

- Bestellmöglichkeit über Webpage seines Fachvereins (sowie bei [spod.ethz.ch](http://spod.ethz.ch)),
- Bequemes Warenkorbsystem
- Beim Bestellen muss ausgewählt werden ob die Bestellung abgeholt wird oder ob eine Postzustellung gewünscht wird
- Postzustellung: nethz-Login → Adresse wird präsentiert (ist vom Kunden änderbar) → Kostenpräsentation (Doks, Porto/Spesen)
- Auswahl/Suche von Dokumenten:
  - Abruf von Sammlungen (durch Fachvereine definiert)
  - Von externen Webseiten soll es möglich sein, durch Anklicken eines direkten Hyperlinks Dokumente automatisch in den Warenkorb zu legen
  - Suche nach Fach/Professor/Prüfungstermin/Studiengang/etc. (individuell nach Studiengang angepasst)
- Kunde wird per E-Mail benachrichtigt wenn die bestellten Dokumente abholbereit sind oder versandt wurden
- Kunde kann Bestellung entweder mit Bargeld vor Ort oder per Rechnung bezahlen (bei Postversand)
- Bestellung kann nach Wunsch durch den Drucker bearbeitet werden:
  - gelocht (2x, 4x)
  - Bostitch
  - Nice to have: Binden, externe Aktionen
- Nice to have: Kunde kann den Status seiner Bestellung anhand einer Auftragsnummer online verfolgen, à la UPS)

### **Gestrichen:**

- Rechnung 'auf die Gasse' → zu hoher Verwaltungsaufwand für SPOD. Dafür Hinweise dass Barzahlung nötig ist.
- Keine nachträgliche Erinnerung, dass bei einer Bestellung die für Abholung markiert ist auch der Postversand möglich ist.

## **Fachvereine / VSETH**

- Admin-Interface: Prüfungssammler loggt sich mit nethz-Account ein (wird von SPOD autorisiert, Whitelist)
- Fachverein bringt neue Dokumente ins System ein:
  1. Vorlesung erstellen
  2. Vorlesung verknüpfen (mit Vorgängervorlesung)
  3. Dokument einfügen
    - Elektronisch (PDF/PS/TIFF/etc.): Upload, Eingabemaske für alle benötigten Infos (inkl. Name/Email/Tel vom Einsender)
    - Papier: Eingabemaske wie oben, Dokument vorbeibringen/schicken (mit ref#). Dokumente ohne ref# werden nicht akzeptiert.
  4. warten, bis Dokumente von SPOD bearbeitet und akzeptiert wurden. erst dann ist es bestellbar.
  5. wenn fertig erhält der Einsender ein Bestätigungsmail

- Verantwortlich für Datenpflege (Studiengangspezifische Informationen und deren richtige Umsetzung im System)
- Verantwortlich für Aktualität der Sammlungen (vordefinierte Dokumentensammlungen, z.B. Basisprüfung 1 der letzten x Jahre)
- Verantwortlich für das Sammeln der Dokumente
- Einlage eines Werbeblattes für Fachvereins-Infos/Events/etc. (nach SPOD-Vorschriften)
- Einbindung der Startseite des Shops in Fachvereins-Webpage, Layout/Bilder durch CSS anpassbar
- Nice to have: Rabatte auf Basis des Departements des Bestellers, entweder pro Bestellung oder pro Kopie

## **Interface für Fachvereine**

- Statistik
- Formular Erfassung neues Dokument und Administration der Daten (Vorlesungen/Dokumente/Infos/etc)
- Preferences (CSS, Werbeblatt, ...)

## **SPOD**

- Interface 'Dokumente': Entgegennahme/Erfassung von Dokumenten
  - PDF-Dateien konvertieren (downsampeln etc), ps2pdf, tiff2pdf, ...
  - Papier: einscannen, Bildkorrekturen
  - In Eingabemaske Angaben nochmals überprüfen und fertiges PDF anhängen und speichern. fertig.
  - Suche nach Dokumenten-ID
  - Sämtliche Informationen/Verlinkungen können durch SPOD verändert werden.
- Verkaufsinterface:
  - Unter Eingabe der Legi-Nr (Barcode auf Legi) liefert System Fachnummer, Preis und weitere Informationen zum Kunden. Verkauf muss dann bestätigt werden.
  - Auflistung aller offenen Bestellungen (noch nicht gedruckt, gedruckt, noch nicht verschickt) mit Details zu Bestellung und Kunde. Hier sollen gewisse Manipulationen möglich sein (erneut drucken, Bestellung temporär aussetzen oder ganz abbrechen)
  - Liste zur Überprüfung der Ablagefächer
  - Interface zu Kasse (Kassenanzeige nach Verkauf, manuelle Einträge, Erinnerung wöchentlicher Kassensturz)
  - Suchfunktion nach Kunden (History)
  - Ausführliche History für alle Bestellungen und Rechnungen
- Interface 'Inhalte' (Möglichkeit: Wordpress CMS)
  - Änderung der Öffnungszeiten
  - Gestaltung der Titelseite des Shops (News einfügen, ...)
- Interface 'Kommunikation'

- Informationen bei Unterbrüchen/Problemen (z.B. an alle Kunden, welche eine Bestellung offen haben)
- Interface 'Admin'
  - Kontrolle über interne Abläufe (z.B. Druckerqueue neu starten, DB-Tests)
  - Abruf Statistiken (Anzahl Bestellungen pro Departement pro Monat/Jahr/etc...)
- Interface 'FIBU'
  - Eingabe ESR-Daten
  - Auflistung aller ausstehenden Zahlungen
  - Möglichkeit, Erinnerungsmail, Mahnungen direkt auszulösen
  - HTML-Ansicht Kassenjournal für Buchhaltung
  - HTML-Ansicht Debitoren für Buchhaltung

## **Anforderungen an die Applikation**

- Dynamische Preise (d.h. Preis auf Knopfdruck anpassbar in der zentralen Konfiguration)
- Möglichkeit für Rabatte (z.B. 1.- Rabatt pro Bestellung)
- Benutzung Authentifizierungssystem ETH/UNI/andere
- Für Kunden, Fachvereine, etc. nethz-Authentifizierung verwenden (Whitelisting)
- Fusszeile mit Dokumentenname, Kundename
- Möglichkeit für weitere Drucker (Zukunft)
- Möglichkeit, Dokumente zu verstecken, dass sie nur über ID bestellbar sind
- Druckerqueue
  - Zeitmanagement
- Möglichkeit Einbindung EC/Postcard
- Intelligenterer Fachmanager der mehrere Bestellungen in einem Fach zusammenlegen kann
- Zentrale Konfigurationsmöglichkeit

## 3 Konzept

### 3.1 Übersicht

Die neue Applikation besteht aus einem Backend und mehreren Frontends.

Das Backend enthält die grundsätzliche Logik des Dokumentenmanagements (siehe Punkt 3.1.1) sowie die Logik des Druckablaufs (siehe Punkt 3.2.2) die den Ausdruck der Bestellungen steuert.

Die Frontends sind alle webbasiert:

- Das Bestell-Interface für die Bestellungen
- Das Fachverein-Interface für das Hinzufügen/Bearbeiten von Dokumenten und Vorlesungen
- Das Administrations-Interface für die Administration des Systems

Diese Frontends sind nicht völlig separat, aber entsprechend den logischen Prozessen aufgeteilt. Das Fachverein-Interface ist demzufolge auch ins Administrations-Interface eingebunden und erlaubt einige zusätzliche administrative Vorgänge die den Fachvereinen vorenthalten bleiben, dies ist in Fig. 1 illustriert.

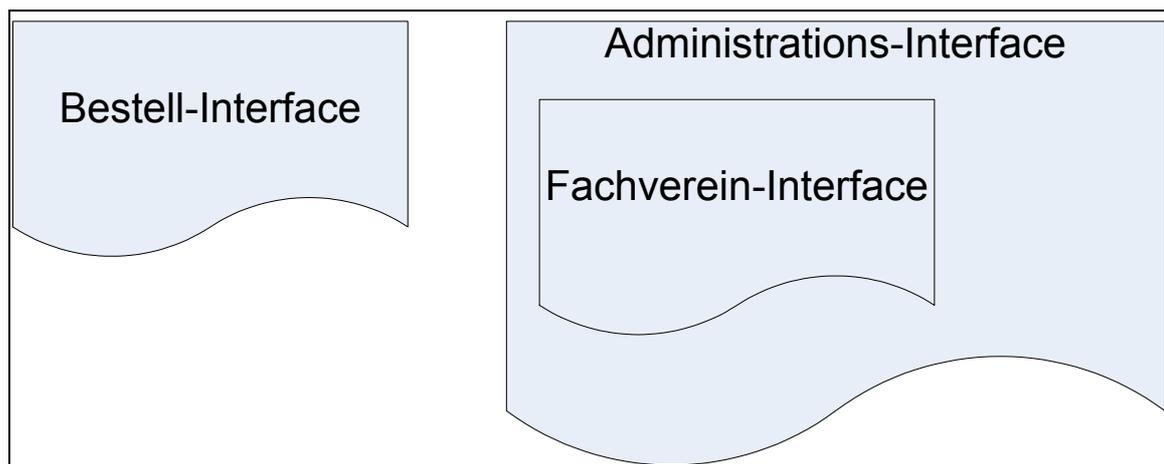


Fig. 1 Die verschiedenen Webinterfaces

Ausserdem gibt es eine zentrale Konfiguration wo globale Optionen, wie z.B. der Standardpreis pro Seite, eingestellt werden können.

An spezifischen Orten im Fachvereinsinterface kommt eine Access Control List (ACL) zum Einsatz. Dies ist notwendig, da einige Elemente sowohl für die Fachvereine wie auch für die SPOD-Mitarbeiter zum Einsatz kommen und damit zum Teil unterschiedliche Berechtigungen bestehen.

#### 3.1.1 Dokumentenmanagement

Die Dokumente werden im Dateisystem abgespeichert. Um Probleme mit Spezialzeichen wie Apostrophe, Schrägstrichen und Leerzeichen zu vermeiden, wird nicht der originale Name sondern die interne ID aus der Datenbank als Bezeichner verwendet.

Das Konzept des Dokumentenmanagements basiert auf miteinander verknüpften Vorlesungen, von denen jede mehrere Dokumente enthalten kann, z.B. Prüfung, Probeprüfung, Lösungen, Übungen, ...

Eine Vorlesung selbst enthält verschiedene Standardattribute wie Name, Jahr, Semester, Professoren, sowie Verknüpfungen mit Departementen.

Daneben definiert jeder Studiengang seine *eigenen* Prüfungsstufen. Dies ist notwendig, da Prüfungsstufen je nach Studiengang verschieden ausfallen können. Bei den einen gibt es noch Vordiplome, bei den anderen nur noch Bachelor-/Masterprüfungen.

Einer Vorlesung können somit auch eine oder mehrere Prüfungsstufen eines oder mehrerer Departemente und Studiengängen zugewiesen werden. Damit wird eine maximale Flexibilität erreicht.

Die Fachvereine des entsprechenden Studiengangs sind zudem dazu angehalten, beim Eingeben von neuen Vorlesungen diese korrekt in die bestehende Dokumentenhierarchie einzusortieren.

Das Ziel dieser Hierarchie ist das automatische Empfehlen von Vorgänger-Vorlesungen zu jeder Vorlesung.

Beispielsweise sollte beim Suchresultat Analysis I, SS 2005 auch die Vorgängervorlesung Analysis I, SS 2004 erscheinen. Damit können frühere Prüfungen und Unterlagen derselben oder einer eng verwandten Vorlesung schnell gefunden werden.

Dafür gibt es drei 'Stärken' von Verknüpfungen; siehe Fig. 2 für ein Beispiel

- **Direkter Vorgänger**  
Bei dieser stärksten Verknüpfung handelt es sich wie im genannten Beispiel um dieselbe Vorlesung, meist ein Jahr zuvor.
- **Indirekter Vorgänger**  
Diese Verknüpfung wird eingesetzt, wenn die ältere Vorlesung kein direkter Vorgänger darstellt sondern z.B. bei einer Vorlesungszusammenlegung nur einen Teilbereich der neueren Vorlesung abdeckt. Dies ist die 'mittelstarke' Verknüpfung da die Vorlesung damit nicht mehr genau abgedeckt wird.
- **Informelle Verknüpfung**  
Dies stellt sozusagen eine nur noch leichte Verknüpfung auf derselben Ebene dar, z.B. um eine eng verwandte Vorlesung zu verknüpfen oder einen optionalen Zusatz zu einer Vorlesung zu definieren.

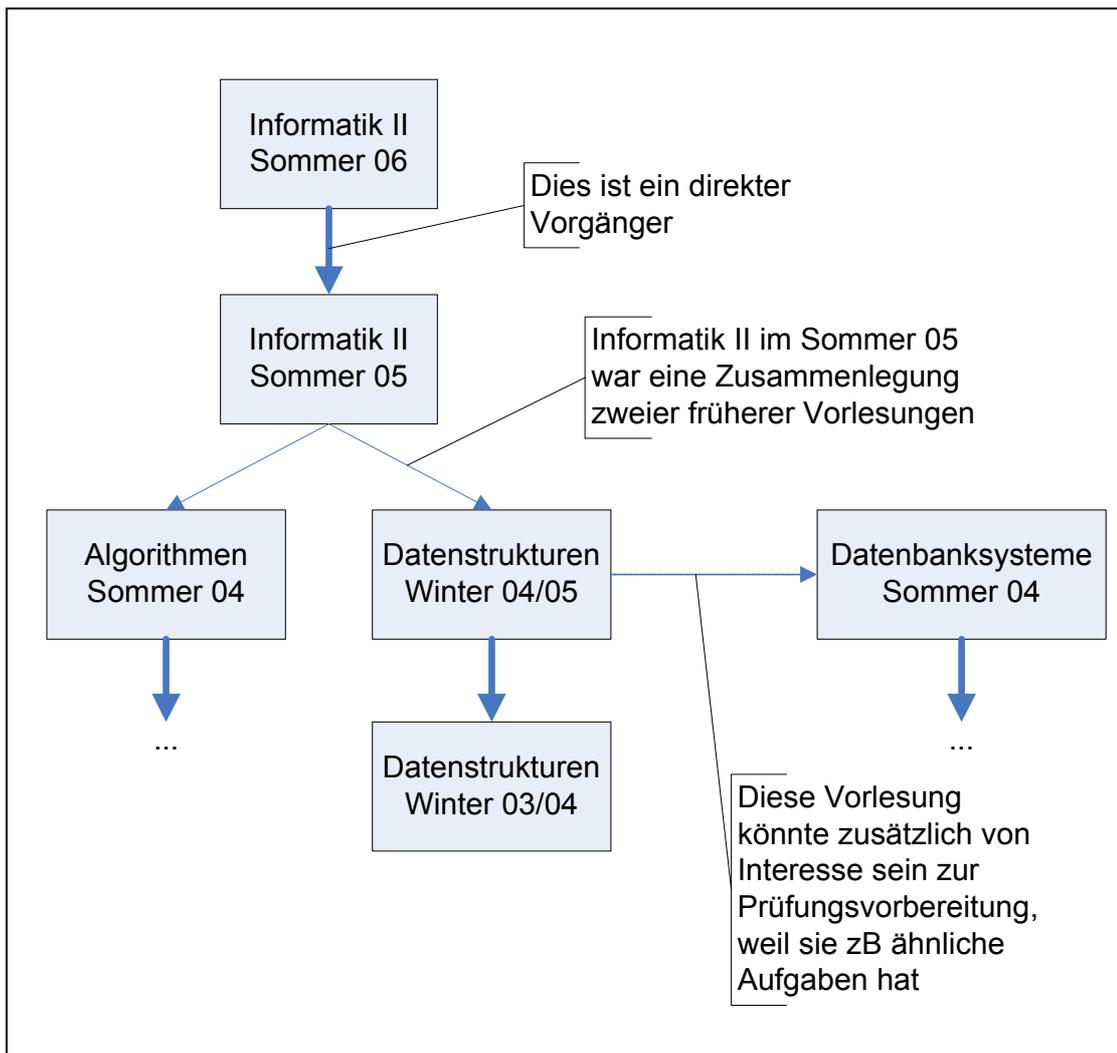


Fig. 2 Fiktives Beispiel einer Hierarchie von Vorlesungen

Der Sinn dieser Klassifizierung liegt vor allem darin festlegen zu können, in wie weit verwandte Vorlesungen aus der Vergangenheit angezeigt werden sollen.

Standardmässig wird nur eine gewisse Anzahl zurückliegender Jahre in Betracht gezogen (z.B. 3; in zentraler Konfiguration einstellbar). Einerseits ist dies, um extrem lange Abhängigkeitsbäume zu vermeiden und andererseits sind lange zurückliegende Dokumente meistens nicht mehr so relevant.

### Der Algorithmus

Der Verknüpfungs-Algorithmus baut eine Baumstruktur der Vorlesungen und ihrer Vorgänger auf. Dieser Baum wird in eine einfach auszuwertende Struktur (z.B. mehrfach verschachteltes Array) eingetragen. Sobald dieser Baum komplett ist, können die relevanten Dokumente angezeigt werden. Ob eine Depth-First oder Breadth-First Strategie zum Zuge kommt sollte damit nicht weiter wichtig sein.

Der Algorithmus zur Suche der anzuzeigenden Dokumente ist wie folgt definiert:

Vorbereitung:

Die bereits über die Suchparameter gefundenen Vorlesungen werden in eine Liste der noch nicht berücksichtigten Vorlesungen ("Trefferliste") eingetragen. Diese kann (und wird) Vorlesungen verschiedenster Jahre enthalten die zum Teil Vorgänger voneinander sind.

Ausgehend von jeder *jeweils jüngsten* Vorlesung werden sämtliche Verknüpfungen überprüft und es wird folgendermassen vorgegangen:

1. **Direkte Vorgänger:**

Vorlesung als Vorgänger in die Struktur eintragen.

Sofern maximale 3 Jahre noch nicht überschritten sind und falls diese Vorlesung in der Trefferliste enthalten ist, diese dort löschen. Daraufhin allen weiteren Verknüpfungen folgen.

2. **Indirekte Vorgänger:**

Vorlesung als Vorgänger in die Struktur eintragen.

Sofern diese Vorlesung nicht in der Trefferliste eingetragen ist fährt der Algorithmus hier nicht mehr weiter.

Falls die Vorlesung jedoch in der Trefferliste eingetragen ist werden von hier erneut alle Verknüpfungen berücksichtigt.

3. **Informelle Verknüpfung:**

Sofern diese Vorlesung nicht in der Trefferliste eingetragen ist, Vorlesung als informelle Verknüpfung in die Struktur eintragen (damit sie etwas gesondert dargestellt werden kann). Darauf fährt der Algorithmus hier nicht mehr weiter.

Falls die Vorlesung jedoch in der Trefferliste eingetragen ist, diese dort belassen, sie wird dann gesondert angezeigt werden als eigenständige Vorlesung da sie ein direkter Treffer war. Damit an dieser Stelle ohne Berücksichtigung dieser Verknüpfung im Algorithmus weiterfahren.

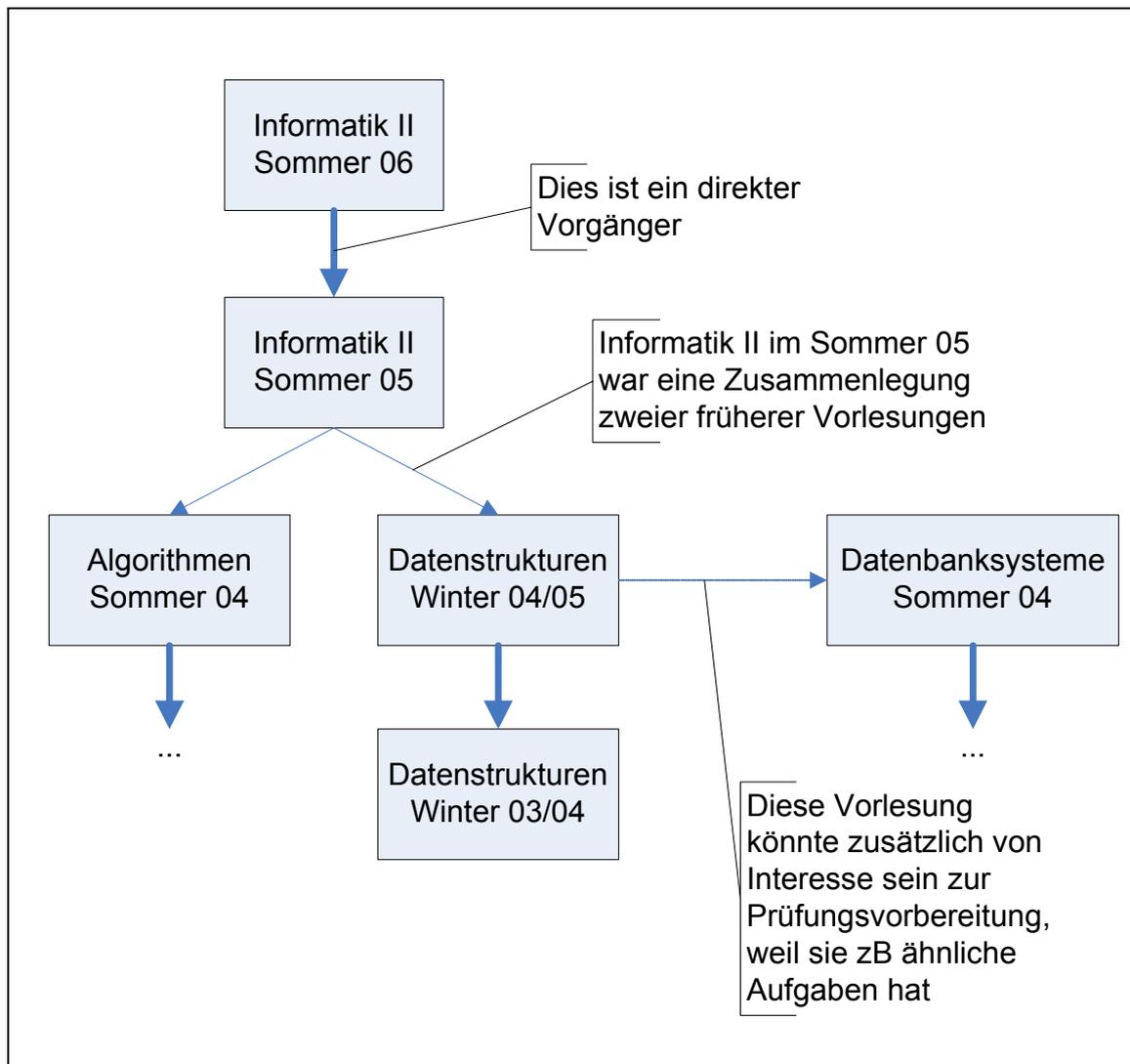


Fig.2 Dieselbe Hierarchie als Illustration für das Beispiel

Das Resultat einer Suche nach "Informatik II Sommer 06" würde also alle Dokumente enthalten von:

- Informatik II Sommer 06
- Informatik II Sommer 05
- Algorithmen Sommer 04
- Datenstrukturen Winter 04/05

### 3.1.2 Bestell-Interface

Das webbasierte Bestell-Interface ermöglicht es den Studierenden, Dokumente zu suchen und eine Bestellung zusammenzustellen (Warenkorb-System). Die fertige Bestellung wird darauf ans Druckablauf-Backend übermittelt.

Ausserdem bietet es einen direkten Einstiegspunkt für vorgängig zusammengestellte Zusammenfassungen über einen Direktlink der die zuvor definierten Dokumente automatisch dem Warenkorb hinzufügt. Jede Sammlung besitzt eine eigene ID im System welche über die URL als Parameter übergeben werden kann.

Das Warenkorb System muss sehr flexibel sein und einen ständigen Zugriff auf dieses gewährleisten um Positionen zu ersehen, zu bearbeiten (Anzahl Ausdrucke, Optionen) und zu entfernen. Der Warenkorb soll stets in einer Kurzfassung ersichtlich sein und auf Knopfdruck auch in einer detaillierten Fassung ersichtlich sein.

Der Ablauf ist vergleichbar mit jedem anderen Onlineshop.

Wenn der Benutzer über die Webseite einsteigt soll er zuerst seine Studienrichtung angeben, damit werden seine Suchen dann sogleich auf die angegebene Studienrichtung eingeschränkt.

Es werden ihm nun die Sammlungen seines Departements angezeigt (sofern vorhanden). Darunter befindet sich eine freie Suchmaske, die eine Suche nach allen Feldern in der Datenbank erlaubt.

Entweder wählt er nun eine vorgefertigte Sammlung an (siehe Fachverein-Interface) worauf alle enthaltenen Dokumente angezeigt werden mit den wichtigsten Informationen (zugehörige Vorlesung mit Dozent, Jahr, Semester und die Beschreibung des Dokuments, Anzahl Seiten und Preis). Entweder kann das Dokument direkt in den Warenkorb gelegt werden oder durch Drücken des zugehörigen Info-Links können sämtliche Details angezeigt werden (in einem Popup-Fenster), die Anzahl bestellter Ausdrucke und auch Spezialoptionen gewählt werden (z.B. lochen, bostitchen, ...)

Oder dann kann die Vorlesungs-Suche verwendet werden welche eine Suche über alle Attribute erlaubt. Felder, welche nicht ausgefüllt werden, gelten als Wildcard.

Die exakten Treffer werden nun in einer Liste angezeigt, absteigend nach Jahr, Semester und (aufsteigend nach) Vorlesungsname sortiert.

Wichtig: In den Resultaten erscheinen *nur* Dokumente die nicht als HIDDEN (versteckt) markiert sind. Diese können *nur* über einen Direktlink gefunden werden. Dies kann z.B. für Skripte verwendet werden.

Für nähere Details siehe das neue Datenbank-Schema im Anhang B.

Zu jeder Vorlesung werden die bereits oben genannten Details angezeigt sowie alle zugehörigen Dokumente der Vorlesung.

Hier gibt es ausserdem den '*verwandte Vorlesungen*' Link. Dabei wird die Datenbank nach verknüpften Vorlesungen durchsucht die nicht jünger als die angewählte Vorlesung sind und zwar maximal 3 Jahre zurück.

Diese Verknüpfung wurde vorgängig durch die Fachvereine definiert (siehe deren Abschnitt) und enthält z.B. die Vorgängervorlesungen oder sonstige ergänzenden und

zugehörigen Vorlesungen an. Dies ist besonders nützlich wenn eine Vorlesung umbenannt wurde oder mehrere Vorlesungen zusammengesetzt wurden und ein eben erst eingetretener Student davon nichts weiss. So kann er diese Vorlesungen ebenfalls finden.

Auch bei diesen Vorlesungen gilt das gleiche Format wie oben mit Anzeige aller Dokumente und dem Info-Link sowie dem Direkt-in-den-Warenkorb-Link

Es soll dazu die Möglichkeit bestehen ein oder mehrere Dokumente über einen Direktlink anzuzeigen ohne dass über eine vordefinierte Sammlung gegangen werden muss. Es wird dann eine Liste angezeigt der definierten Dokumente die sogleich bestellt werden können. Jedes Dokument besitzt eine eigene ID im System welche über die URL als Parameter übergeben werden kann.

Der Kunde kann jederzeit den Warenkorb einsehen wo er die aktuell markierten Dokumente, deren Preis sowie den Gesamtpreis sehen kann.

Auch hier hat er Zugriff auf den Info-Link zur Detailanzeige (genau wie in der Auflistung zuvor) sowie die Möglichkeit eine Position aus dem Warenkorb zu entfernen.

Beim klicken auf den Knopf zur Bestellauslösung muss er sich nun anmelden am System (via n.ethz Login resp. AAI).

Es wird nun noch einmal eine definitive Zusammenfassung der Bestellung angezeigt mit dem Gesamtpreis. Ausserdem kann via Drop Down eine Postzustellung geordert werden. Der Preis wird automatisch berechnet und als weitere Position angezeigt. Der Gesamtpreis aktualisiert.

Die Versandkosten werden zentral definiert mit dem Kosten pro Couvert (deckt Porto, Couvert, Rechnungsformular, Arbeit) sowie die Anzahl Seiten pro Couvert (Damit das Couvert als normaler Brief versendet werden kann)

Entweder kann nun wieder zurückgegangen werden in die Suchmaske (wenn etwas vergessen wurde, wenn Positionen korrigiert werden müssen etc) oder dann kann die Bestellung ausgelöst werden.

Wenn Abholung ausgewählt wurde (Standard), wird die Bestellung im System ausgelöst und als abzuholen markiert. Der Kunde erhält eine Bestätigungsseite.

Wird Postzustellung gewünscht, wird seine eingetragene Adresse angezeigt (z.B. ausgelesen als Attribut aus AAI), die er jedoch auch noch modifizieren kann sollte er eine andere Lieferadresse wünschen.

Nach der Bestätigung der Lieferadresse wird die Bestellung im System ausgelöst und der Kunde erhält eine Bestätigungsseite.

### 3.1.3 Fachvereins-Interface

Ein Fachverein ist

- verantwortlich für das Sammeln der Dokumente
- verantwortlich für Datenpflege (Studiengangspezifische Informationen und deren richtige Umsetzung im System, z.B. Definition der Prüfungsstufen)
- verantwortlich für Aktualität der Sammlungen (vordefinierte Dokusammlungen, z.B. Basisprüfung 1 der letzten x Jahre)

Jeder Fachverein kann einen Verantwortlichen bestimmen, der für die jeweilige Dokumenten-Suche und –Erfassung zuständig ist. Dieser wird vom SPOD in die ACL eingetragen und hat fortan Zugriff auf dieses Interface.

Zentral ist eine ACL (Access Control List) definiert (siehe weiter vorne), welche Rechte an Benutzer (Leginummer; nur diese ist eindeutig) knüpfen kann. Somit werden nur eingeschulten Leuten die Rechte gegeben, Dokumente einzubringen, Vorlesungen zu modifizieren, zu verknüpfen etc.

Bevor jemand jedoch diese Rechte bekommt wird er von SPOD eingeschult. Ihm werden die relevanten Teile des Systems erklärt und demonstriert. Sobald er diese Punkte verstanden hat werden ihm die Berechtigungen gewährt.

SPOD Mitarbeiter greifen ebenfalls direkt auf dieses Interface zu, wenn sie Dokumente freigeben oder bearbeiten etc müssen.

- Admin-Interface: ein Prüfungssammler loggt sich mit seinem nethz-Account ein (wird von SPOD via eine Whitelist autorisiert, siehe ACL)
- Fachverein bringt neue Dokumente ins System ein:
  1. Vorlesung erstellen
  2. Vorlesung verknüpfen (mit Vorgängervorlesung)
  3. Dokument einfügen
    - Elektronisch (PDF/PS/TIFF/...): Upload, Eingabemaske für alle benötigten Infos (inkl. Name/E-Mail/Tel vom Einsender)
    - Papier: Eingabemaske wie oben, Dokument vorbeibringen/schicken (mit ref# der Eingabe). Dokumente ohne ref# werden nicht akzeptiert!
  4. warten, bis Dokumente von SPOD bearbeitet und akzeptiert wurden. erst dann ist es bestellbar. Dieser Schritt ist notwendig um die Qualität der eingegebenen Daten zu gewährleisten.
  5. wenn fertig erhält der Einsender ein Bestätigungsmail
- Einlage eines Werbeblattes für Fachvereins-Infos/Events/etc. (nach SPOD-Vorschriften)
- Einbindung der Startseite des Shops in FV-Webpage, Layout/Bilder durch CSS anpassbar

## **Interface**

Die Rechte sind jeweils in der ACL gespeichert! Nicht jeder Benutzer hat damit zwangsläufig Berechtigung auf alle dieser Funktionen:

- Statistiken
- Erstellen/Modifizieren von Vorlesungen
- Verknüpfen von Vorlesungen
- Erfassung neuer Dokumente und Administration der Daten
- Preferences (CSS, Upload Werbeblatt, News, ...)
- Nur für SPOD Mitarbeiter: Bearbeiten und Freigeben der Vorlesungsdokumente

### 3.1.4 Administrations-Interface

Auf dieses Interface haben nur SPOD-Mitarbeiter Zugriff. Es ist das umfangreichste Interface, welches sowohl den Zugriff auf das Fachverein-Interface erlaubt um Vorlesungen und Dokumente zu Administrieren wie auch Zugriff auf Funktionen erlaubt für die Bestellungsabwicklung, der Kasse etc.

- Interface ‘Dokumente’: Entgegennahme/Erfassung von Dokumenten
  1. PDF-Dateien manuell konvertieren (downsampeln etc), ps2pdf, tiff2pdf, ...
  2. Papier: einscannen, Bildkorrekturen
  3. In Eingabemaske Angaben nochmals überprüfen und fertiges PDF anhängen und speichern.
  4. Probedruck starten
    - Suche nach File-ID
    - Sämtliche Informationen/Verlinkungen können durch SPOD verändert werden.
- Verkaufsinterface:
  - Unter Eingabe der Leginr (Barcode auf Legi) liefert System Fachnummer, Preis und weitere Informationen zum Kunden. Verkauf muss dann bestätigt werden.
  - Auflistung aller offenen Bestellungen (noch nicht gedruckt, gedruckt, noch nicht verschickt) mit Details zu Bestellung und Kunde. Hier sollen gewisse Manipulationen möglich sein (neu drucken, auf ‘temp’ setzen, ...)
  - Liste zur Überprüfung der Ablagefächer
  - Interface zu Kasse (Kassenanzeige nach Verkauf, manuelle Einträge, Erinnerung wöchentlicher Kassensturz)
  - Suchfunktion nach Kunden (History)
  - Ausführliche History für alle Bestellungen und Rechnungen
- Interface ‘Admin’
  - Kontrolle über interne Abläufe (z.B. Druckerqueue neustarten, DB-Tests, was weiss ich...)
  - Bearbeiten der ACL für das Fachverein-Interface
  - Abruf Statistiken (Anzahl Bestellungen pro Dept pro Monat/Jahr/etc...)
- Interface ‘FIBU’
  - Eingabe ESR-Daten
  - Auflistung aller ausstehenden Zahlungen
  - Möglichkeit, Erinnerungsmail, Mahnungen direkt auszulösen
  - Export Kassenjournal für Buchhaltung
  - Export Debitoren für Buchhaltung
- Interface ‘Reservationssystem’
  - Fachvereine wie auch SPOD können Druckzeiten reservieren.
- Interface ‘Kommunikation’
  - Infos bei Unterbrüchen/Problemen (z.B. an alle die Bestellung offen haben)
- Interface ‘Inhalte’ (Möglichkeit: Wordpress CMS <sup>6</sup>)

- Änderung Öffnungszeiten
- Gestaltung der Titelseite des Shops (News einfügen, ...)

## **3.2 Module**

### **3.2.1 Login**

Bisher wird der Login über das n.ethz System vorgenommen was jedoch den Nutzerkreis auf die ETH limitiert.

Um auch Studenten anderer Universitäten und Hochschulen den Zugang zum System zu gewähren sollte ein übergreifendes Login-System verwendet werden wie AAI<sup>7</sup> welches das authentifizieren an einer beliebigen dem System angeschlossenen Universität erlaubt (derzeit 11 Universitäten sowie 6 weitere in Vorbereitung).

Dies kann jedoch auch erst in einer zweiten Stufe gemacht werden. Doch viele Unis tendieren derzeit dazu den direkten LDAP Zugriff nicht mehr zu gewähren und die Benutzung von AAI zu verlangen.

Switch AAI funktioniert über ein Apache Modul 'Shibboleth Framework' und kann ziemlich einfach eingebunden werden (Apache 1.3.x und 2.x).

Genaue Installationsanweisungen für Linux und Mac OS X finden sich auf der AAI Homepage.

## Prinzip der AAI Authentifikation:

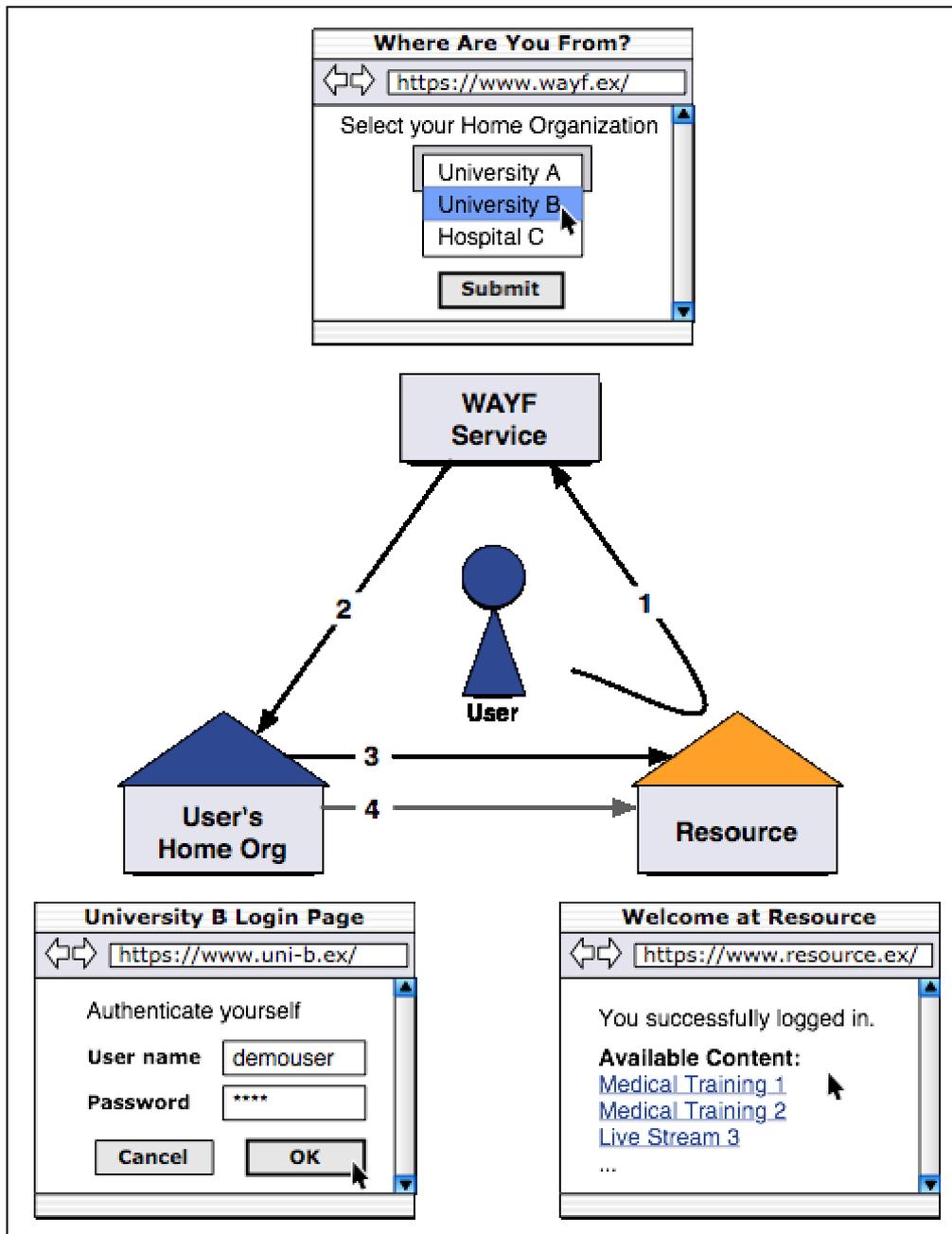


Fig. 3 AAI Prinzip <sup>8</sup>

1. Wenn der Benutzer eine geschützte Seite anfordert und er nicht bereits zuvor eingeloggt war wird er zuerst zum WAYF-Service umgeleitet ('Where Are You From'-Service). Hier muss zuerst die Heim-Universität ausgewählt werden.
2. Darauf wird der Benutzer auf die heimische Loginseite seiner Universität umgeleitet wo er sich mit seinen Benutzerdaten anmelden kann.
3. Der Benutzer wird nach erfolgreicher Anmeldung wieder auf die ursprüngliche Seite geleitet wo er nun auf den geschützten Inhalt zugreifen kann.
4. Im Hintergrund werden ausserdem einige Benutzerdaten und zusätzliche Attribute übertragen die die Seite benötigt (z.B. die Postadresse, Departement, ...)

Ein Benutzer kann später wiedererkannt werden über seine Unique ID von AAI. Diese ist garantiert über alle angeschlossenen Universitäten eindeutig und enthält keinerlei Daten die auf den Benutzer schliessen lassen (insbesondere nicht die Legi-Nummer).

In der aktuell gültigen Spezifikation ist jedoch leider die Legi-Nummer nicht enthalten aus Datenschutzgründen. Da jedoch immer mehr Anwendungen auch Zugriff auf diese Daten brauchen sieht der aktuelle Draft dieses Attribut ebenfalls vor.

Zumindest die Uni Zürich hat sich bereits mit dem Problem beschäftigt und plant, dieses Attribut bald einzuführen. Von der ETH sind ebenfalls erste Willensbekundungen bekannt; sobald das Projekt jedoch konkret aufgegleist wird muss bei beiden beteiligten Hochschulen noch etwas Druck ausgeübt werden damit diese Attribute ebenfalls freigegeben werden. Dies sollte laut Auskunft von AAI jedoch kein grosses Problem darstellen mit diesem konkreten use-case.

Die neue Spezifikation die noch in der Entwicklung ist findet sich unter *Referenzen/AAI\_Attr\_Specs\_v1\_1\_5.pdf*

Danach kann dann die Legi-Nummer genau wie andere Attribute wie die Adresse etc bei Bedarf von AAI angefordert werden.

Da das System vorläufig sowieso auf die ETH Zürich und Uni Zürich limitiert ist stellt dies kein Problem dar. Mit dem Release dieses neuen Drafts sollten dann auch die anderen Universitäten früher oder später nachziehen.

### **3.2.2 Druckablauf**

Sobald eine Bestellung ausgelöst wurde wird die Komponente des Druckablaufs aktiv. Diese ist dafür zuständig dass die Bestellung ordnungsgemäss ausgedruckt wird und ggf. der Versand ausgelöst wird. Dieser Ablauf wird in Fig. 5 aufgezeigt.

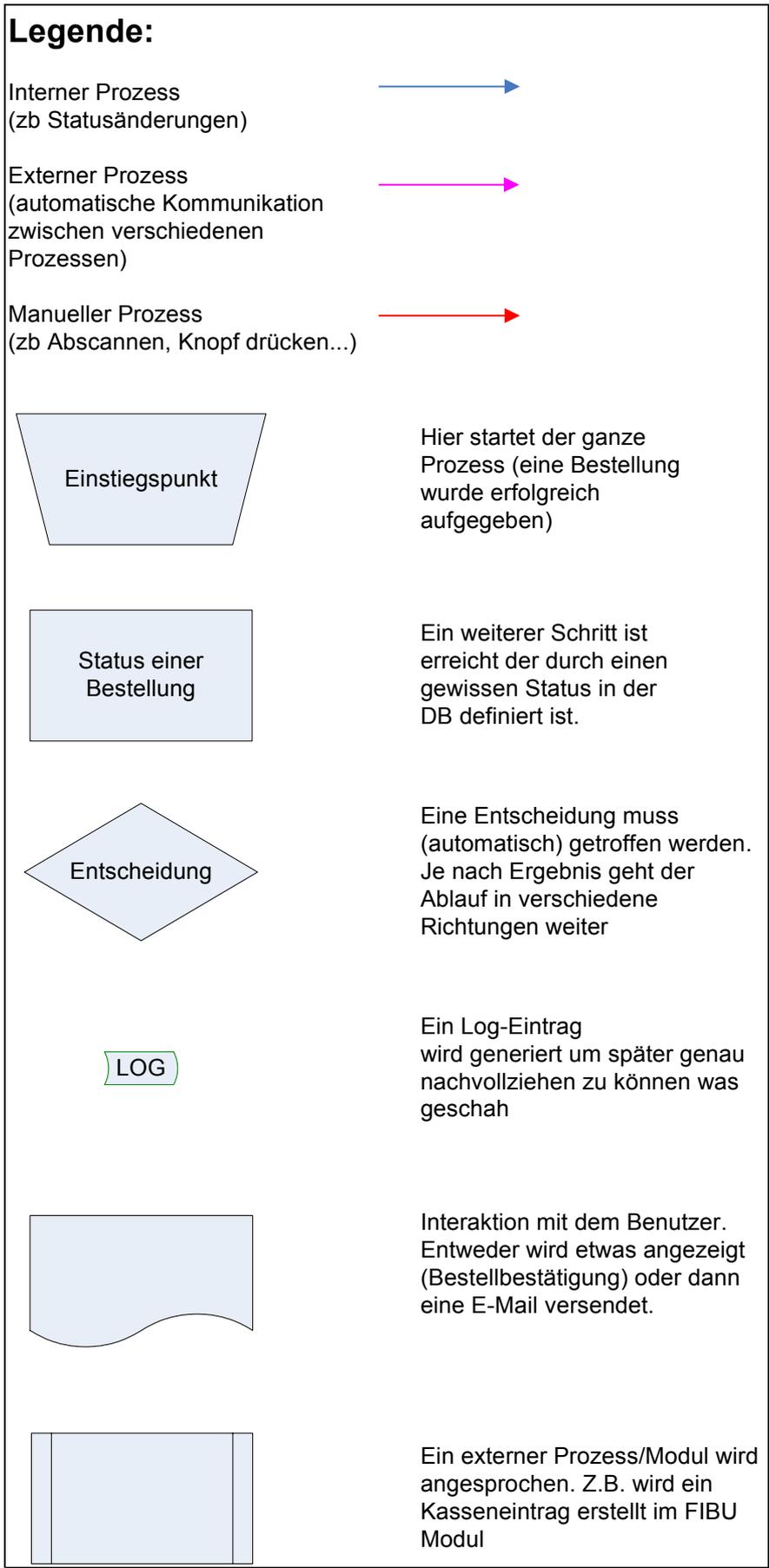


Fig. 4 Legende zum Druckablauf

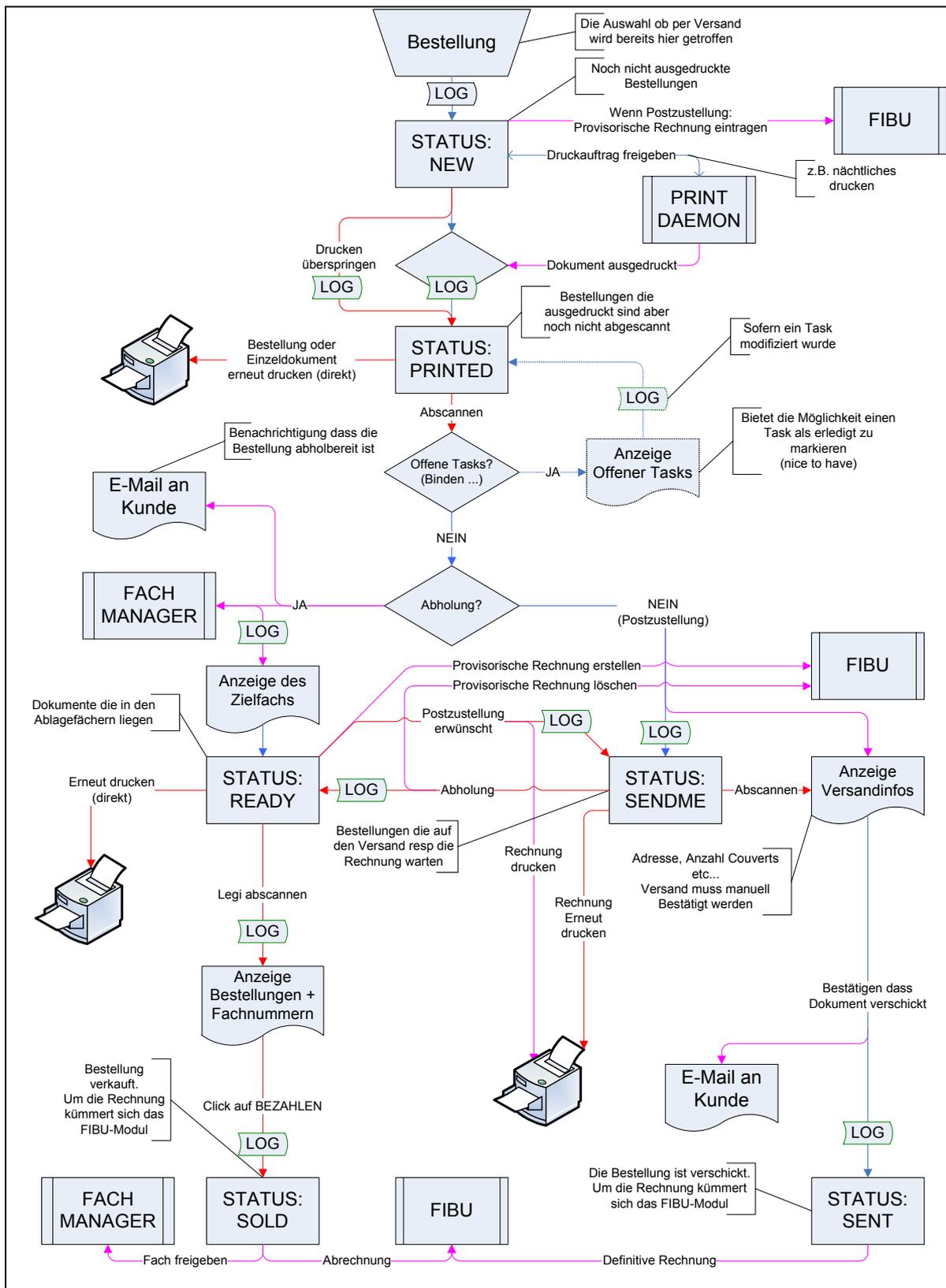


Fig. 5 Interner Ablauf des Druckablaufs

Bei allen manuellen Vorgängen **muss** der aktuelle Status überprüft werden damit keine Inkonsistenzen entstehen.

Jede Statusänderung oder manuelle Aktion wird bei der Bestellung in der Datenbank protokolliert mit Zeit, User (wo möglich) und Aktion

## Globale Optionen:

Es besteht die Möglichkeit einen Auftrag an jeglicher Stelle anzuhalten indem ein Flag bei der Bestellung in der Datenbank gesetzt wird, z.B. *active=0* . Damit können keine weiteren Aktionen auf diesem Auftrag unternommen werden (weder automatische noch manuelle) bis er wieder aktiv geschaltet wurde oder *gelöscht* wird

## Legende:

- A** Automatischer Prozess (z.B. Statusänderungen) (→)
- E** Externer Prozess (automatische Kommunikation zwischen verschiedenen Prozessen via IPC oder Sockets) (→)
- M** Manueller Prozess (z.B. Abscannen, Knopf drücken...) (→)

## STATUS: NEW

Die Bestellung wurde gerade erst aufgegeben; der Status der Bestellung wird als NEW vermerkt.

Der Printdaemon druckt im Allgemeinen nur nachts oder zu vorgegebenen Zeiten also bleibt die Bestellung vorerst im System. Dies ist so vorgesehen um die Ausdrücke zu kanalisieren und Nachts ist der Drucker im Allgemeinen nicht durch andere Aufgaben belegt.

- M** Über das Administrationsinterface kann der Druck auch übersprungen werden (z.B. weil bereits ein vorgedrucktes Exemplar vorliegt) womit der Status auf PRINTED gesetzt wird.
- E** Wenn der Printdaemon die Bestellung fertig gedruckt hat (inklusive eventueller Rechnung) wird der Status auf PRINTED gesetzt. Bei Druckerproblemen wie Papierstau oder Ausgabefach voll pausiert der Drucker automatisch und wartet auf die manuelle Behebung. Dies blockiert automatisch auch den Printdaemon der bei Behebung des Problems dann fortfährt.

## STATUS: PRINTED

Die ausgedruckten Bestellungen müssen nun beim Drucker abgeholt werden. Pro Auftrag wird ein Titelblatt gedruckt. Der Barcode auf dem Titelblatt enthält die ID des Auftrags, welche mit dem Handscanner gescannt wird.

- M** Wenn die Bestellung oder ein Einzeldokument z.B. fehlerhaft gedruckt wurde kann entweder die gesamte Bestellung oder ein Einzeldokument noch einmal direkt ausgedruckt werden.
- M** Wenn ein Titelblatt abgescannt wird wird geprüft, ob noch offene Aufgaben wie z.B. Binden erledigt werden müssen bei allen oder einigen Dokumenten.

Es wird eine Liste der Tasks angezeigt die jeweils mit einem Haken versetzt werden müssen wenn er erledigt wurde. Auch diese Aktion wird geloggt inklusive dem Benutzer der die Aktion bestätigte. Derzeit ist nur 'binden' vorgesehen, in Zukunft können hier aber weitere Aufgaben hinzukommen.

Wenn entweder keine Tasks vorhanden sind oder alle erledigt sind wird geprüft ob ein Versand gewünscht wurde:

- A** Abholung: der Fachmanager wird angefragt, in welches Fach das Dokument abgelegt werden soll. Diese Nummer wird angezeigt und der Status des Auftrags wird auf READY gesetzt. Damit ist das Dokument abholbereit.

Der Kunde wird per E-Mail informiert dass die Bestellung abholbereit ist.

- A** Versand: Die Lieferadresse und die Anzahl Couverts wird angezeigt; siehe Status SENDME für Details. Entweder wird das Dokument gleich verpackt oder kann auf einen Stapel gelegt werden, wenn gerade keine Zeit dafür vorhanden ist.

Der Auftragsstatus wird ausserdem auf SENDME gesetzt. Das Dokument liegt bereit für den Versand.

### **Status: READY**

Die Bestellung ist abholbereit.

- M** Die gesamte Bestellung oder einzelne Dokumente können bei Bedarf erneut gedruckt werden (direkt an den Drucker), z.B. wenn nun noch Druckerfehler entdeckt wurden oder ein Dokument verloren ging.

- M** Postzustellung erwünscht: Auf Wunsch kann das Dokument dennoch verschickt werden. Damit wird eine provisorische Rechnung in die FIBU eingetragen und diese sofort auf ESR-Papier ausgedruckt (Dieses Papier befindet sich in einem konfigurierbaren Fach im Drucker). Die Dokumente müssen nun *sofort* verpackt werden oder zumindest zusammen mit der Rechnung auf dem Stapel landen.

- M** Wenn der Abholer die Nummer auf seiner Legi abscannen lässt resp. über die Suchfunktion nach dem Namen gefunden wurde werden seine offenen Bestellungen angezeigt.

Diejenigen Bestellungen welche READY sind und bei welchen Abholung gewünscht wurde werden zusammengefasst und die Fachnummer(n) der Dokumente sowie der Gesamtpreis wird angezeigt.

Es besteht nun auch die Möglichkeit den Preis zu korrigieren als positiven oder negativen Korrekturbetrag. Wenn dies verwendet wird muss jedoch *zwingend* das Kommentarfeld dazu ausgefüllt werden. Kann verwendet werden für Rabatte bei schlechtem Ausdruck, Zuschlag z.B. für eine Schachtel etc...

Der Mitarbeiter hat nun zwei Möglichkeiten (z.B. per Checkbox auswählbar):

- M** Barzahlung:

Der Status ändert sich auf SOLD und dem FIBU-Modul wird eine Barzahlung übermittelt

Dem Fachmanager wird die Freigabe der betroffenen Fächer übermittelt

### **Status: SOLD**

Die Bestellung wurde abgeholt und ist erledigt. Dies besagt jedoch noch nichts darüber ob eine mögliche Rechnung bezahlt wurde. Darum kümmert sich unabhängig davon das FIBU-Modul.

**Status: SENDME**

Die Bestellung ist fertig ausgedruckt und aus dem Drucker entfernt, muss aber noch verpackt werden für den Versand.

Sowohl beim ersten abscannen nach dem Drucken wie auch beim späteren abscannen vom Stapel wird der Empfänger sowie die Anzahl Couverts angezeigt.

**M** Sobald die Bestellung verpackt ist und bereit liegt für den Versand muss eine entsprechende Bestätigung angewählt werden. Damit wechselt der Status zu SENT und in der FIBU wird die Rechnung als definitiv eingetragen.

**M** Rechnung erneut drucken: Damit wird die Rechnung erneut direkt ausgedruckt, wenn sie z.B. verloren ging.

**M** Abholung: Wenn z.B. per Telefon oder persönlichem Erscheinen dennoch eine Abholung gewünscht wurde *und die Bestellung noch nicht den Status SENT hat* kann dies noch rückgängig gemacht werden. Damit wechselt der Status auf READY (für die Abholung) und die provisorische Rechnung wird aus der FIBU gelöscht.

**Status: SENT**

Damit gilt die Bestellung als versendet und die Rechnung als definitiv. Dies kann *nicht* mehr geändert werden. Dies ist aus so gelöst um das System einfach zu halten. Seltene Sonderfälle mit Rücksendungen, Falschlieferungen etc müssen gegebenenfalls manuell erledigt werden.

### 3.2.3 Reservationssystem

Das Reservationssystem wird von der Druckersteuerung zur Regulierung der Druckerbenutzung verwendet. Wichtig ist hier dass die zukünftige Möglichkeit mehrerer Drucker berücksichtigt wird. Jeder Drucker besitzt dann seine eigenen Reservationen. Eine Reservation ist also immer einem Drucker zugewiesen und nicht global.

Im Allgemeinen werden die angefallenen Bestellungen über Nacht ausgedruckt, so dass sie am nächsten Tag bereit zur Einsortierung oder zum Versand bereit sind. Damit können die Druckaufträge kanalisiert werden (der Drucker braucht lange zum Aufwärmen) und die Leerlaufzeit nachts ideal genutzt werden.

Manchmal gibt es jedoch auch Ausnahmen oder Änderungen die zentral definiert werden können. Zum Beispiel kann es sein dass in einer Nacht einmal Broschüren oder andere Dokumente in grosser Auflage gedruckt werden, dann sollten nicht plötzlich Bestellungen dazwischen kommen.

Hierfür ist das Reservationssystem gedacht wo mit verschiedenen Prioritätsleveln gearbeitet wird. Je höher das Level, desto höher die Priorität:

Level 0:

Das Standardlevel welches per Default überall gilt wo sonst nichts definiert ist. Dieses Level kann keinem Eintrag zugewiesen werden.

Level 1:

Das zweittiefste Level welches von allen anderen übersteuert wird. Dieses kann z.B. anderen Fachvereinen zugewiesen werden, die die Auflage haben den Drucker für alle anderen Tasks jederzeit freizugeben.

Level 2:

Das normale SPOD-Hintergrunddruck-Level. Also das Level das z.B. über Nacht definiert wird wenn die Druckersteuerung freie Hand hat die aufgelaufenen Bestellungen abzuarbeiten.

Level 3:

Für wichtige Reservationen wie z.B. ein weiteres Druckfenster gleich vor der Präsenzzeit kann diese zweithöchste Priorität zugewiesen werden.

Level 4:

Dies ist die höchste Priorität die z.B. benutzt werden kann um Sperrzeiten zu definieren. Wenn gerade eine wichtige Sitzung läuft dann soll der Drucker während dieser Zeit keinen unnötigen Lärm verursachen, daher kann hier ein Fenster definiert werden wo nichts automatisch gedruckt wird.

Natürlich verhindert dieses Level keinen manuellen Druckauftrag, was auch Sinn der Sache ist – schliesslich muss auch mal bei einer Sitzung kurz etwas gedruckt werden.

### 3.2.4 Printdaemon

Da der momentane Drucker XEROX 4550 nur einen Windows und einen Mac OS-X Treiber besitzt, ist hier eine Mac OS-X Station nötig die den ganzen Druck übernimmt. Zum Einsatz kommt hier CUPS<sup>7</sup>, das Standard-Drucksystem in Mac OS X sowie den meisten Linux Derivaten. Mac OS-X verwendet eine leicht angepasste Version von CUPS dessen Kommandozeilensyntax jedoch identisch ist.

Darauf soll ein Daemon erstellt werden der einerseits Zugriff zum Reservationssystem besitzt wie auch auf die Daten und Datenbanken der Hauptapplikation zugreifen kann.

Da in Zukunft auch mehrere Drucker vorhanden sein können die ihre eigenen Einträge im Reservationssystem aufweisen muss dieser Ablauf für jeden Drucker parallel ablaufen können! (Multi Threading oder Multi-Processing)

Wichtig für den Daemon ist vor allem die volle Kontrolle über die Queue. Das heisst der Daemon muss jederzeit den Status des Druckauftrages einsehen können. Da der Daemon mit dem Reservationssystem gekoppelt ist, kann er nicht einfach nur seine Aufträge 'blind' an den Drucker schicken (wie beim VPP) sondern kann nur immer einen Auftrag aufs Mal abarbeiten. Nach Beendigung dessen muss erneut kontrolliert werden ob weitergedruckt werden darf.

Der interne Ablauf pro Drucker ist in Fig. 6 dargestellt.

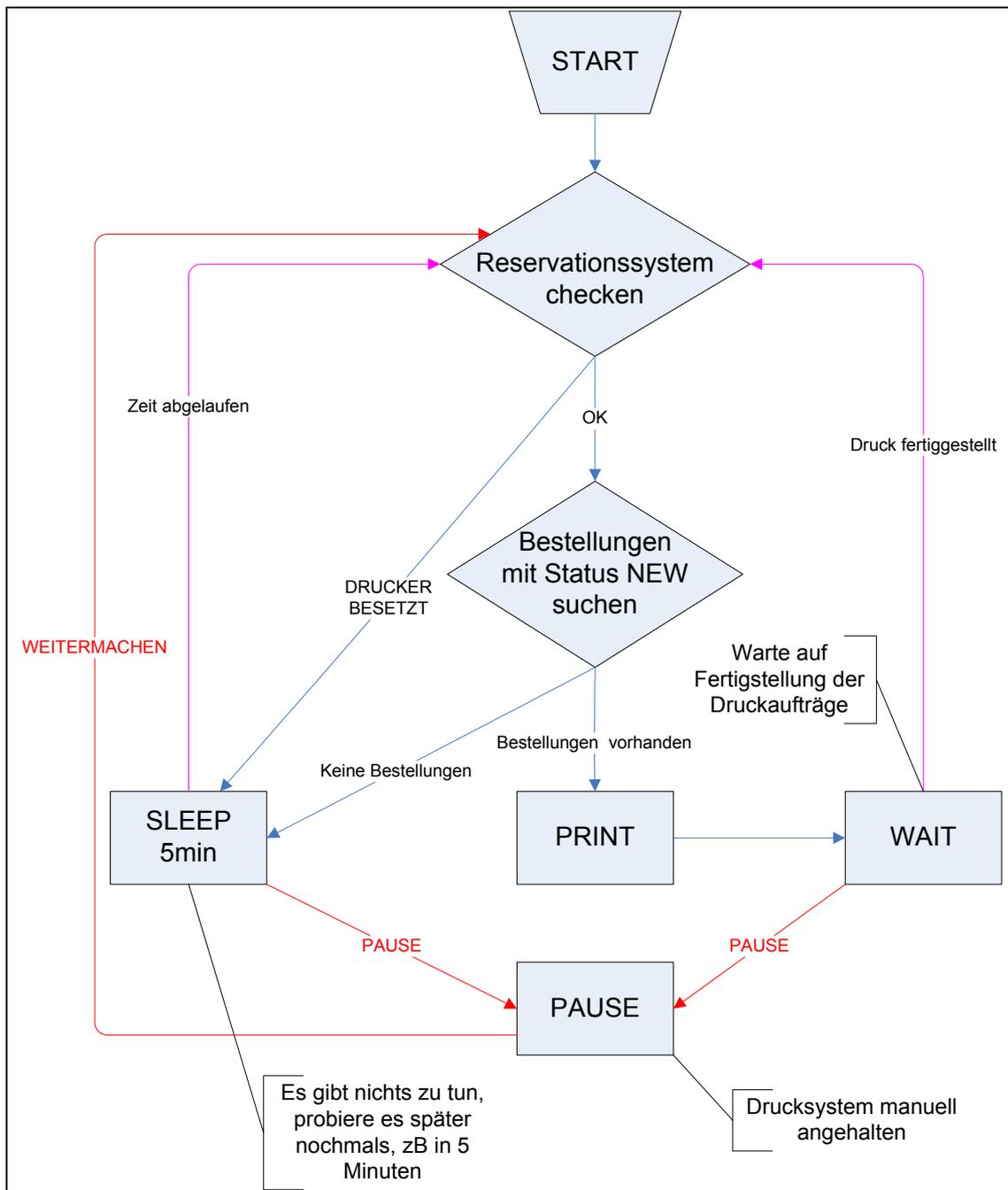


Fig. 6 Diagramm der Druckerlogik

Beim ersten Ausführen oder beim späteren Wiedereinstieg wird zuerst das Reservationssystem angefragt ob der Daemon momentan überhaupt drucken darf. wenn nein, soll der Daemon 5 Minuten warten und dann erneut beginnen.

Wenn er grundsätzlich drucken darf, muss er auch noch überprüfen, ob es überhaupt etwas zu drucken gibt in der Datenbank (Datensätze die den Status NEW haben und nicht inaktiv sind). Falls es keine gibt soll der Daemon 5 Minuten warten und dann erneut beginnen.

Da die Zeit in der gedruckt werden darf u.U. relativ kurz sein kann wird nur immer eine Bestellung aufs Mal gedruckt und dann erneut überprüft ob weiterhin gedruckt werden darf.

Ein Druckauftrag besteht aus:

- Titelseite mit Barcode (Anderes Papier)
- Rechnung (Spezielles Rechnungspapier)
- Falls Versand + grosser Ausdruck Rechnungskopie(n), nummeriert!
- Falls definiert: Werbeblatt des Departements des Bestellers (nur 1 Seite)
- Bestellte Dokumente

Die Titelseite, Rechnung und ggf. Rechnungskopien werden dynamisch vom System erzeugt und dem Daemon fertig als Dateien übergeben.

Das Werbeblatt (sofern vorhanden) ist ebenfalls bereits als PDF Datei im System gespeichert (über das Fachvereinsinterface) und kann auch direkt übergeben werden.

Die einzelnen Dokumente können dann via **lp** an CUPS übergeben werden.

Der Daemon soll nun solange warten bis **lpq -P dest** keinen offenen Auftrag mehr anzeigt. Dann kann er von vorne beginnen und ggf. weitere Bestellungen ausdrucken.

Der fertig ausgedruckte Auftrag wird ausserdem im System als PRINTED markiert sodass er nicht erneut gedruckt wird.

Ausserdem gibt es eine manuelle Aktion, den Daemon jederzeit zu stoppen.

Wenn dieser sich gerade im SLEEP befindet beginnt er nicht nach 5 Minuten mit einem neuen Durchlauf sondern wartet bis der wieder manuell gestartet wird.

Wenn er sich im Druck befindet werden alle offenen Druckaufträge gelöscht mittels **lprm -P dest** – (der Strich am Schluss wird benötigt um *alle* Aufträge zu löschen).

Erst wenn der Daemon manuell wieder aufgefordert wird, weiterzumachen beginnt er erneut am Anfang des Loops.

### 3.2.5 Fachmanager

Der Fachmanager sorgt für die Zuteilung einer gedruckten Bestellung in die vorhandenen Fächer.

Das Prinzip des neuen Fachmanagers ist relativ einfach; eine flache Datenbankstruktur ist hier wohl sinnvoll:

In der Datenbank ist jedes Fach aufgelistet mit folgenden Angaben:

- Fach-ID (die auch am physischen Fach angeschrieben ist)
- Maximale Anzahl Seiten die das Fach bietet
- Besteller (Leginr (sofern vorhanden) und UniqueID des Bestellers)
- Bestellungen (die IDs der Bestellungen die da ggf. drin liegen)
- Anzahl Seiten (von allen Bestellungen – kann auch näherungsweise nur die Anzahl Seiten der Dokumente sein; ohne Titelseiten, Werbeblätter, ...)

Durch die Speicherung der Bestellungen-IDs kann einerseits bei einer möglichen Inkonsistenz zurückverfolgt werden was in einem Fach lag. Ferner dient die Information als Hinweis für jene Person, welche die Seiten aushändigt dass es mehrere Bestellungen sind.

Dieses Modul besitzt folgende interne Schnittstellen, über welche andere Module auf die Dienste zugreifen kann:

- **Eine Liste aller Fächer mit ggf. vorhandenen Bestellungen ausgeben**  
Dies dient zur manuellen Fachkontrolle zur Erkennung von Inkonsistenzen
- **Einer fertig ausgedruckten Bestellung ein Fach zuweisen**  
Beim Abscannen nach dem Ausdruck sagt diese Routine wo die Seiten abgelegt werden sollen. Dieses Interface wird jedoch bei Postzusendung übergangen da hier kein Fach benötigt wird!  
Anhand der Legi-Nr./ID wird gesucht ob dieser Besteller bereits fertige Bestellungen in einem oder mehreren Fächern hat. Falls ja wird überprüft ob der Platz noch für die aktuelle Bestellung reicht worauf dieses Fach dann wiederverwendet werden kann. Wird jedoch kein Fach gefunden oder der Platz reicht nicht mehr wird ein neues Fach zugewiesen welches gross genug für diese Bestellung ist.  
Sollten sämtliche Fächer belegt sein wird ein bereits belegtes Fach welches noch genug Platz für die aktuelle Bestellung aufweist.
- **Anhand einer Legi-Nr. alle Fächer anzeigen die Bestellungen dieses Benutzers enthalten**  
Da beim Abholen mehrere Bestellungen zusammen abgerechnet werden, muss auch alles zusammen ausgehändigt werden. Falls der Platz für ein Fach nicht gereicht hat müssen daher ggf. alle Fachnummern angezeigt werden wo Bestellungen dieses Benutzers liegen.  
Sollte ein Fach wegen Platzmangels von mehreren Benutzern belegt sein wird eine

Warnung für dieses Fach angezeigt die auf diesen Umstand hinweist.

- **Anhand einer Fachnummer alle Bestellungen darin anzeigen**  
Zur schnellen Kontrolle eines Faches
  
- **Anhand einer Bestellnummer das zugehörige Fach suchen**  
Zur schnellen Lokalisierung einer Bestellung
  
- **Ein Fach als leer markieren (wenn die Dokumente abgeholt sind)**  
Nach erfolgter Abholung der Bestellung(en)

### 3.2.6 FIBU Modul

Das FIBU Modul ist für die Finanzverwaltung zuständig und verwaltet sowohl die lokal vorhandene Kasse wie auch alle Rechnungen.

Einträge werden entweder bei erfolgreichem Barverkauf, dem Einlesen der ESR-Daten oder bei einem manuellen Eintrag erstellt. All diese Aktionen können nur von SPOD-Mitarbeitern getätigt werden.

Es gibt vorerst 3 verschiedene Typen von Einträgen in die Finanzbuchhaltung:

#### 1. Barzahlung (Kasse) (Automatischer Eintrag beim Verkauf)

Wenn etwas Bar verkauft wurde im Shop wird dieser Eintrag verwendet. Er enthält auch die IDs der verkauften Dokumente.

Zu diesem Typ zählt auch eine möglicherweise zukünftige Bezahlung per EC/Postcard/...

#### 2. Barbezug/Einzahlung (Kasse) (Manueller Eintrag)

Für sonstige mit Bargeld bezahlte Geldbewegungen wie Einkauf von Materialien und dergleichen

#### 3. Rechnung (Konto)

Wenn etwas per Rechnung verkauft/verschickt wurde. Dieser Eintrag enthält mehr Angaben als die Barzahlung für die Nachverfolgung des Bezahlstatus. Er enthält u.A. das Datum der Erfassung, das Datum und den Status der letzten Aktion (Offen, Zahlungserinnerung, Mahnung1, Mahnung2) sowie das Datum der Bezahlung. Ebenso soll ein Kommentarfeld vorhanden sein um eine Nachricht zu hinterlegen in Spezialfällen mit Mahnungen oder Betreibungen.

Jede Rechnung enthält eine Referenznummer (VESR) die auf dem Einzahlungsschein abgedruckt wird. Diese besteht aus der Rechnungsnummer sowie einer Prüfziffer nach modulo 10 rekursiv.

Der Standardzustand ist OFFEN, d.h. die Rechnung ist noch nicht bezahlt. Wenn die Rechnung bezahlt wird ändert er auf BEZAHLT (kann auch durch ein BOOLEAN Wert in der Datenbank realisiert werden). Eine teilweise bezahlte Rechnung gilt noch immer als OFFEN. Hier ist ein manueller Eingriff nötig um z.B. fehlendes Geld nachzufordern.

Dazu gibt es den aktuellen Status der wie folgt definiert ist: (Siehe auch weiter unten 'Mahnungen')

- Neue Rechnung: <leer>
- Nach 30 Tagen sobald das automatische Zahlungserinnerungs-E-Mail geschrieben wurde: ERINNERUNG

- Nach weiteren 2 Wochen werden an einem (in der zentralen Konfiguration) definierten Wochentag die Mahnungen gedruckt ändert der Status auf MAHNUNG1
- Einen Monat nach der ersten Mahnung wird analog zum vorigen Punkt eine weitere Mahnung gedruckt und der Status ändert auf MAHNUNG2
- Wird auch die zweite Mahnung nicht bezahlt ist manuelle Bearbeitung erforderlich sollte tatsächlich eine Betreuung oder ähnliches nötig sein.

### **ESR:**

Das FIBU-Modul besitzt ausserdem eine Schnittstelle für Bankdaten (ESR). Mit dem regelmässigen Einlesen der Daten können offene Rechnungen automatisch als bezahlt markiert werden.

Dies funktioniert über die Referenznummer welche auch mit jedem Auftrag gespeichert wird. Beim Einlesen der Bankdaten wird damit jede Rechnung die offen ist und mit einer Referenznummer übereinstimmt als bezahlt markiert und der ESR-Eintrag zur Kontrolle in die Datenbank eingetragen.

Die technischen Details zu ESR sind im ESR-Kapitel unter *Implementierung* erläutert.

### **Mahnungen:**

In periodischen Abständen kann das Modul (per Cron-Job) die offenen Rechnungen überprüfen und nach dem ersten Monat eine Zahlungserinnerung per E-Mail schreiben.

Jeden Montag Morgen sollten ausserdem automatisch offene Mahnungen gedruckt werden (in einen anderen Tray als die sonstigen Bestellungen!) und zwar:

Nach 1½ Monaten eine erste Mahnung und nach 2½ Monaten eine zweite Mahnung.

Ein Administrator kann ausserdem die Liste der offenen Rechnungen einsehen und den aktuellen Bezahlstatus erkennen (siehe Beschreibung zum SPOD-Interface).

### **Kontrolle:**

Alle Aktionen, seien sie automatisch (ESR, Rechnungseröffnung vom Hauptmodul, ..) oder manuell (Barbezug, ..), werden im FIBU-Eintrag protokolliert so dass später ein möglichst genaues Nachvollziehen des Ablaufes möglich ist wenn Probleme oder Unstimmigkeiten auftreten.

### **Schnittstelle:**

Das FIBU-Modul wird vom Druckablauf-Modul oft benutzt und muss diese Funktionalität anbieten:

Provisorische Rechnung erstellen

Wenn bei der Bestellung oder später bei einer manuellen Umschreibung ein Postversand gewünscht wurde muss eine provisorische Rechnung erstellt werden. Diese wird letztendlich mit dem Versand definitiv.

#### Provisorische Rechnung löschen

Wenn eine Bestellung abgebrochen (gelöscht) oder in eine Abholung umgewandelt wird, wird die provisorische Rechnung aus dem System entfernt.

#### Provisorische Rechnung fixieren

Sobald die Bestellung versendet wurde wird die Rechnung definitiv und muss bezahlt werden.

#### Barzahlung eintragen

Trägt bei Barzahlung den bezahlten Betrag in die Kasse ein.

## 4 Implementierung

### 4.1 Printersteuerung

Als Druckserver wurde CUPS<sup>9</sup> ausgewählt da es sehr flexibel ist und man Druckaufträge gut nachverfolgen kann, ob ein Auftrag noch aussteht oder bereits gedruckt ist. Wichtige Punkte waren der vorhandene Treiber, die generelle Flexibilität mit verschiedensten Formaten umzugehen und schliesslich auch dass CUPS eigentlich in jedem Linux automatisch mit dabei ist.

Es gibt keinen offiziellen Linux-Treiber für den aktuellen Drucker XEROX 4110, doch es gibt einen Mac OS-X Treiber, welches ein etwas verändertes CUPS-System einsetzt, dessen Interface und Befehle jedoch dasselbe ist.

Das spezielle an diesem Drucker ist jedoch dass er ein internes Accounting besitzt benötigt er einen Benutzernamen und ein Passwort welches normalerweise über das Druckertreiber-GUI eingegeben wird.

Leider sind diese Einstellungen jedoch aus der Kommandozeile *nicht* verfügbar.

Die Lösung besteht darin den Xerox-eigenen Filter welcher nach der Konvertierung zu einem Postscript ausgeführt wird

Der Filter befindet sich unter:

/Library/Printers/Xerox/Filter/XRPSAFHFilter

In CUPS wird der Filter mit 5 oder 6 Argumenten aufgerufen die folgendermassen definiert sind:

<p><b>filtername</b> <i>printer job user title num-copies options [filename]</i></p> <p><i>printer</i> Der Name der printer queue (normalerweise ist das der Name des Programms)</p> <p><i>job</i> Die numerische job-ID</p> <p><i>user</i> Der String des <i>originating-user-name</i> Attributes</p> <p><i>title</i> Der String des <i>job-name</i> Attributes</p> <p><i>copies</i> Der numerische Wert des <i>number-copies</i> Attributes</p> <p><i>options</i> String Repräsentation der Job-Attribute, getrennt durch Leerzeichen. Angaben als "name=wert" for Einzelwertattribute und "name=wert1,wert2,...,wertN" für Mehrfachattribute</p> <p><i>filename</i> Optional: Die Datei (wird bei dieser Verwendung nicht benutzt)</p>
---

Fig. 7 Auszug aus dem CUPS Software Programmers Manual<sup>10</sup>

Da dieser Filtername nicht in CUPS selbst verändert werden kann, kann hier nur ein Wrapper zum Einsatz kommen der anstelle des Originalfilters eingesetzt wird (mit dessen Name). Der Originalfilter wird umbenannt.

Der Wrapper ruft dann zuerst den Originalfilter auf und bearbeitet dann dessen Ausgabe. Die Parameter können direkt 'durchgereicht' werden und brauchen nicht weiter ausgewertet zu werden.

Letztendlich soll es das bearbeitete Postscript-File auf STDOUT schreiben.

Ein einfaches Proof-of-Concept Wrapper-Script ist:

```
#!/bin/sh

/Library/Printers/Xerox/Filter/XRPSAFHFilter_ "$1" "$2" "$3" "$4" \
"$5" | sed '1,2d' | cat /path/to/new/header -
```

Da der Filter bereits zwei Header einsetzt werden die hier einfach entfernt.

Die ersten paar Zeilen *vor* dem "%!PS-Adobe-3.0" müssen sein:

```
ESC%-12345X@PJL JOB MODE=PRINTER
@PJL SET JOBATTR="@JOAU=<username>"
@PJL SET JOBATTR="@JOEP=<password>"
@PJL SET JOBATTR="@DAID="
@PJL SET JOBATTR="@ACNA=WBS Ausdruck"
@PJL ENTER LANGUAGE=POSTSCRIPT
%!PS-Adobe-3.0
```

"ESC" steht für das Zeichen mit dem ASCII Code 0x1B

Die ersten zwei Zeilen des normalen Outputs von XRPSAFHFilter können auch ermittelt werden durch direkten Aufruf und etwas beliebigen Text. Dann werden diese 2 Zeilen auch hinzugefügt die durch die obigen ersetzt werden müssen. (Es ist die erste und die sechste Zeile)

<username> ist der verwendete Username für den Drucker

<password> ist das verschlüsselte Passwort. Da die Verschlüsselung nicht dokumentiert ist, ist die verschlüsselte Form am einfachsten zu ermitteln durch einen Druckauftrag aus dem GUI heraus wo der entsprechende Eintrag dann herausgesniff wird oder durch eben diesen Wrapper in eine Datei geschrieben wird. Dann kann daraus das verschlüsselte Passwort ausgelesen werden.

Sobald diese Informationen am Ende des Filters vorhanden sind kann der Drucker die Datei ausdrucken.

**WICHTIG:** Eine Kopie des Wrapperscripts separat speichern da es ansonsten bei einem Treiberupdate möglicherweise überschrieben wird !

## 4.2 ESR (E-Banking)

Als Bank wurde die ZKB gewählt welche ein Online-Banking System unterhält und erlaubt ESR-Daten herunterzuladen wo die relevanten Kontobewegungen festgehalten sind. Daten werden im ESR TYPE 3-Format übermittelt.

Das File besteht aus mehreren Gutschrift-Records welche am Schluss mit einem Totalrecord abgeschlossen sind. Pro Record wird eine Zeile verwendet.

Die genaue Dokumentation findet sich in folgender Datei:

*Referenzen/vesr\_handbuch.pdf*

Hier die wichtigsten Daten:

Einzelne Zahlungen werden als *Gutschriftrecord* gespeichert. Fig. 8 zeigt den Aufbau eines Gutschriftrecords

Gutschriftrecord (Typ 3)							Anhang
Feld-Nr.	Feldname	Zeichen Positionen	Format <sup>1</sup>	Picture (COBOL-ANS)	Inhalt	Abhängigkeit/Bemerkungen	
1	2	3	4	5	6	7	
1	Transaktionsart	1-3	3 N	X(3)	Siehe Ziffer 2.6		
2	Teilnehmernummer	4-12	9 N	9(9)	99999999P P - Prüfziffer (Modulo 10, rekursiv)	01-200027-2	
3	Referenznummer (inkl. Prüfziffer)	13-39	27 N	9(27)	1. Kundenid.-Nummer 2. Referenz Kunde 3. Prüfziffer	6 Stellen 20 Stellen 1 Stelle	
4	Betrag	40-49	10 N	9(8)V99			
5	Aufgabereferenz	50-59	10 AN	X(10)	Bank-interne Angaben	Leerzeichen an unterschiedlichen Positionen	
6	Aufgabedatum	60-65	6 N	9(6)	JJMMTT JJ - Jahr MM - Monat TT - Tag	Einzahlungsdatum bei Post- oder Bankstelle bzw. Überweisungsdatum durch Checkamt oder Bank/belegloser Zahlungsverkehr	
7	Verarbeitungsdatum	66-71	6 N	9(6)	JJMMTT JJ - Jahr MM - Monat TT - Tag	Einlesedatum im Rechenzentrum der Postfinance oder der Bank	
8	Gutschriftsdatum	72-77	6 N	9(6)	JJMMTT JJ - Jahr MM - Monat TT - Tag	Datum der Gutschrift auf das Bankkonto/ Buchungsdatum ZKB	
9	Mikrofilm-Nr./Ref.	78-86	9 N	9(9)		Für Nachforschungen unerlässlich	
10	Rejectcode inkl. Prüfziffer	87	1 N	9		0 - nicht rejectiert 1 - rejectiert auf Zeile 21 5 - Massenreject	
11	Reserve	88-96	9 AN	X(9)	NULL («0» Nullen)		
12	Einzahlungstaxe	97-100	4 N	99999		siehe Beilage	

<sup>1</sup> N = numerisch, AN = alphanumerisch

Fig. 8 Gutschriftrecord in ESR

Am Ende der Datei befindet sich ein *Totalrecord*. Dieser dient zur Kontrolle der Gutschriftrecords und ist eine Zusammenfassung dieser und enthält unter Anderem die Anzahl Gutschriftrecords sowie die Gesamtsumme. Fig. 9 zeigt den Aufbau eines Totalrecords

Totalrecord (Typ 3)							Anhang
Feld-Nr.	Feldname	Zeichen Positionen	Format <sup>1</sup>	Picture (COBOL-ANS)	Inhalt	Abhängigkeit/Bemerkungen	
1	2	3	4	5	6	7	
1	Transaktionsart	1-3	3 N	X(3)	999	Siehe Ziffer 2.6	
2	Teilnehmernummer	4-12	9 N	9(9)	99999999P P = Prüfziffer (Modulo 10, rekursiv)	01-200027-2	
3	Sortierschlüssel (inkl. Prüfziffer)	13-39	27 N	X(27)	1. Kundenid.-Nummer 2. Füll-Neuner 9.....9	11- oder 12-stellige Kontonummer 27 Stellen	
4	Betrag	40-51	12 N	9(10)V99			
5	Anzahl Transaktionen	52-63	12 N	9(12)		ohne TOTALRECORD	
6	Erstellungsdatum	64-69	6 N	9(6)	JJMMTT JJ - Jahr MM - Monat TT - Tag	Erstellungsdatum des Datenträgers im Rechenzentrum	
7	Einzahlungstaxen	70-78	9 N	9(7)V99		Total Postspesen für Einzahlungen am Postschalter	
8	Reserve	79-87	9 N	9(7)V99	00000000	Nullen	
9	Reserve	88-100	13 AN	X(13)	Spaces	Nullen	

<sup>1</sup> N = numerisch, AN = alphanumerisch

**Fig. 9 Totalrecord in ESR**

Jeder Rechnungseintrag in der Datenbank hat die Referenznummer bereits gespeichert und ist daher einfach aufzufinden.

Das ESR-File wird zeilenweise eingelesen und die Referenznummer gesucht.

Wenn die Rechnung bereits als bezahlt markiert wurde muss nichts weiter getan werden (vermutlich wurde dann dieser Record bereits früher einmal eingelesen und sollte daher nicht erneut berücksichtigt werden)

Wenn die Rechnung noch nicht bezahlt wurde wird der Rechnungsbetrag mit dem Zahlungsbetrag verglichen.

Bei Übereinstimmung wird die Rechnung als bezahlt markiert und der gesamte Record in der Datenbank gespeichert um eine mögliche spätere Rekonstruktion zu vereinfachen.

Bei Fehleingaben der ESR-Nummer eines Kunden ist manuelle Aktion nötig um die korrekte Rechnung zu finden und um diese dann als bezahlt zu markieren.

## 5 Evaluation und Ausblick

In Zusammenarbeit mit diversen SPOD Mitarbeitern wurden die Anforderungen erstellt, diskutiert und klassifiziert.

Das in dieser Arbeit dargestellte System erfüllt alle Anforderungen die als wichtig betrachtet werden für einen ersten Release.

Einige Features wurden als 'nice to have' markiert. Dies sind Features die nicht im ersten Release sein müssen aber für eine weitere Version in Betracht gezogen werden. Es handelt sich hier natürlich nicht um kritische Punkte sondern nur um kosmetische oder bedienerfreundlichere Erweiterungen.

Andere Features wurden noch während der Evaluation wieder verworfen (z.B. die 'Rechnung auf die Gasse' da beschlossen wurde dass dies mehr Aufwand (im Betrieb) bedeutet als Nutzen bringt und wohl nie implementiert werden wird).

Derzeit wird ein neues Team rekrutiert für die eigentliche Implementation des Systems

### **Aktueller, grober Zeitplan:**

- Ende März 2006 : Kostenabschätzung
- Anfangs April 2006: Werbung (in Polykum, VIS-Foren etc.)
- Organisation Meeting mit interessierten Programmierern
- Koordination mit Janick Bernet (Projektleitung)
- Ende April 2006: Beginn Neuprogrammierung

## 6 Schlussfolgerungen

Die Applikation ist ziemlich komplex und besteht aus vielen Teilen die reibungslos miteinander kooperieren müssen. Dies sicherzustellen ist nicht einfach und es muss gut programmiert werden.

Es war interessant so eine ziemlich grosse Applikation zu planen wenn auch nicht immer einfach, da viele verschiedene Leute miteinbezogen waren und dies manchmal auch zu einigen Verzögerungen geführt hat.

Gerade dies macht für den Unerfahrenen eine genaue Zeitplanung ziemlich schwierig und ich habe mich hier auch öfters verschätzt.

Dennoch war die Arbeit sehr interessant und herausfordernd eine Lösung zu finden die allen Anforderungen entspricht und möglichst nichts verbaut für zukünftige Anforderungen.

# A Altes Datenbank-Schema

Database: wbs\_dokudb Table: ablage Rows: 121

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned		PRI	0		select,insert,update,references
md	timestamp(14)		YES			select,insert,update,references
pages	int(10) unsigned		0			select,insert,update,references
orders	int(10) unsigned		0			select,insert,update,references
maxpages	int(10) unsigned			500		select,insert,update,references

Table | Non unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment |

ablage	0		PRIMARY	1		A			121		BTREE	
--------	---	--	---------	---	--	---	--	--	-----	--	-------	--

Database: wbs\_dokudb Table: admins Rows: 33

Field	Type	Null	Key	Default	Extra	Privileges
id	int(11)				auto_increment	select,insert,update,references
name	varchar(100)					select,insert,update,references
vornname	varchar(100)					select,insert,update,references
email	varchar(100)					select,insert,update,references
status	enum('active','inactive')			YES		select,insert,update,references
pathlimit	varchar(200)			YES		select,insert,update,references
md	timestamp(14)			YES		select,insert,update,references
username	varchar(12)			YES		select,insert,update,references
password	varchar(16)			YES		select,insert,update,references
zustaeendig	set('main','docs','upload','kasse','all')			YES		select,insert,update,references
zustaendig	varchar(100)			YES		select,insert,update,references

Table | Non unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment |

admins	0		PRIMARY	1		A			33		BTREE	
--------	---	--	---------	---	--	---	--	--	----	--	-------	--

Database: wbs\_dokudb Table: banners Rows: 4

Field	Type	Null	Key	Default	Extra	Privileges
id	int(11)				auto_increment	select,insert,update,references
bannerpath	varchar(255)		YES			select,insert,update,references
kunde	varchar(255)		YES			select,insert,update,references
hits	int(10) unsigned		YES	0		select,insert,update,references
studiengaenge	varchar(255)					select,insert,update,references
status	enum('inactive','active')		YES	active		select,insert,update,references
link	varchar(255)					select,insert,update,references

Table | Non unique | Key\_name | Seq\_in\_index | Column\_name | Collation | Cardinality | Sub\_part | Packed | Null | Index\_type | Comment |

banners	0		PRIMARY	1		A			4		BTREE	
---------	---	--	---------	---	--	---	--	--	---	--	-------	--

Database: wbs\_dokudb Table: barcodes Rows: 1

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned		PRI		auto_increment	select,insert,update,references
code	varchar(10)	YES				select,insert,update,references
bezeichnung	varchar(100)	YES				select,insert,update,references
verkauft	int(10) unsigned	YES				select,insert,update,references
preis	double(8,2)	YES		0.00		select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references

Table	Non unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
barcodes	0	PRIMARY	1	id	A	1				BTREE	

Database: wbs\_dokudb Table: bestellung Rows: 11226

Field	Type	Null	Key	Default	Extra	Privileges
best_id	int(10) unsigned		PRI		auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
cd	timestamp(14)	YES				select,insert,update,references
k_id	int(11)	YES	MUL	0		select,insert,update,references
status	enum('temp','new','in progress','printed','mailed','done','sent')	YES	MUL	temp		select,insert,update,references
optionen	text	YES				select,insert,update,references
sbid	varchar(32)	YES	UNI			select,insert,update,references
bestellt	timestamp(14)	YES	MUL	0000000000000000		select,insert,update,references
gedruckt	timestamp(14)	YES		0000000000000000		select,insert,update,references
benachrichtigt	timestamp(14)	YES		0000000000000000		select,insert,update,references
remote	varchar(15)					select,insert,update,references
ablage_fach	int(10) unsigned			0		select,insert,update,references
heften	enum('false','true')			false		select,insert,update,references
duplex	enum('false','true')			true		select,insert,update,references
lastclick	varchar(32)					select,insert,update,references
lochen	int(11)			0		select,insert,update,references

Table	Non unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
bestellung	0	PRIMARY	1	best_id	A	11295				BTREE	
bestellung	0		1	sbid	A	11295				BTREE	
bestellung	1		1	k_id	A	11295				BTREE	
bestellung	1		1	bestellt	A	11295				BTREE	
bestellung	1		1	status	A	2		YES		BTREE	

Database: wbs\_dokudb Table: copyrights Rows: 1

Field	Type	Null	Key	Default	Extra	Privileges
id	int(11)		PRI		auto_increment	select,insert,update,references
text	varchar(100)	YES				select,insert,update,references
pathlimit	varchar(200)	YES				select,insert,update,references

Table	Non unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
copyrights	0	PRIMARY	1	id	A	1				BTREE	

Database: wbs\_dokudb Table: dokument Rows: 2347

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned				auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
cd	timestamp(14)	YES		00000000000000		select,insert,update,references
psfile	varchar(200)		UNI			select,insert,update,references
seiten	int(10) unsigned	YES				select,insert,update,references
docu_md	timestamp(14)	YES		00000000000000		select,insert,update,references
zusatz	varchar(50)	YES				select,insert,update,references
art	varchar(100)					select,insert,update,references
status	enum('prov','def','old')	YES		prov		select,insert,update,references
typ	enum('Prüfung','Skript','Übung','Buch')			Prüfung		select,insert,update,references
version	int(10) unsigned			2		select,insert,update,references
mailed	int(10) unsigned	YES		0		select,insert,update,references

Table	Non unique	Key name	Seq_in_index	Column name	Collation	Cardinality	Sub_part	Packed	Null	Index type	Comment
dokument	0	PRIMARY	1	id	A	2323				BTREE	
dokument	0	psfile_2	1	psfile	A	2323				BTREE	
dokument	1	psfile_1	1	psfile	A	2323				BTREE	

Database: wbs\_dokudb Table: enthaelt Rows: 229884

Field	Type	Null	Key	Default	Extra	Privileges
dok_id	int(11)				MUL	select,insert,update,references
best_id	int(11)				MUL	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references

Table	Non unique	Key name	Seq_in_index	Column name	Collation	Cardinality	Sub_part	Packed	Null	Index type	Comment
enthaelt	1	i_enthaelt_1	1	best_id	A	21912				BTREE	
enthaelt	1	i_enthaelt_2	1	dok_id	A	1585				BTREE	

Database: wbs\_dokudb Wildcard: gehalten\_von

Tables
gehalten_von

Database: wbs\_dokudb Wildcard: gehoert\_zu

Tables
gehoert_zu

Database: wbs\_dokudb Wildcard: gnuccash\_konten

Tables
gnuccash_konten

Database: wbs\_dokudb Table: kasse Rows: 10085

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned		PRI		auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
datum	date	YES				select,insert,update,references
text	varchar(100)	YES				select,insert,update,references
admin	varchar(20)	YES				select,insert,update,references
betrag	double(8,2)	YES				select,insert,update,references
soll	varchar(4)					select,insert,update,references
haben	varchar(4)					select,insert,update,references
art	enum('Bl1','Bl2','Bl3')	YES				select,insert,update,references

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
kasse	0	PRIMARY	1	id	A	5271				BTREE	

Database: wbs\_dokudb Table: kontaktperson Rows: 0

Field	Type	Null	Key	Default	Extra	Privileges
dok_id	int(11)		PRI	0		select,insert,update,references
prof_id	int(11)		PRI	0		select,insert,update,references

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
kontaktperson	0	PRIMARY	1	dok_id	A	0				BTREE	
kontaktperson	0	PRIMARY	2	prof_id	A	0				BTREE	
kontaktperson	1		1	dok_id	A	0				BTREE	
kontaktperson	1		2	prof_id	A	0				BTREE	

Database: wbs\_dokudb Table: kunde Rows: 4894

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned		PRI		auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
name	varchar(50)					select,insert,update,references
vorname	varchar(50)	YES				select,insert,update,references
strasse	varchar(8)	YES				select,insert,update,references
plz	varchar(30)	YES				select,insert,update,references
ort	varchar(100)	YES				select,insert,update,references
email	varchar(8)	YES				select,insert,update,references
legi	varchar(50)	YES				select,insert,update,references
password	varchar(32)	YES	MUL			select,insert,update,references
authentication	enum('wbs','n.ethz')	YES		wbs		select,insert,update,references
blocked	enum('false','true')	YES		false		select,insert,update,references
mailings	int(10) unsigned	YES	MUL	1		select,insert,update,references

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
kunde	0	PRIMARY	1	id	A	4854				BTREE	
kunde	1		1	username	A	4854			YES	BTREE	
kunde	1		1	mailings	A	2			YES	BTREE	

Database: wbs\_dokudb Table: log Rows: 26793

Field	Type	Null	Key	Default	Extra	Privileges
id	int(11)				auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
ts	timestamp(14)	YES		00000000000000		select,insert,update,references
best_id	int(11)			0		select,insert,update,references
k_id	int(11)			0		select,insert,update,references
text	varchar(255)	YES				select,insert,update,references
r_id	int(11)	YES		0		select,insert,update,references

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
log	0	PRIMARY	1	id	A	26557				BTREE	

Database: wbs\_dokudb Table: oeffnungszeiten Rows: 146

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned				auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
datum	date	YES				select,insert,update,references
zeit	varchar(20)	YES				select,insert,update,references
admin	varchar(15)	YES				select,insert,update,references

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
oeffnungszeiten	0	PRIMARY	1	id	A	146				BTREE	

Database: wbs\_dokudb Table: professor Rows: 1679

Field	Type	Null	Key	Default	Extra	Privileges
prof_id	int(11)				auto_increment	select,insert,update,references
md	timestamp(14)	YES				select,insert,update,references
name	varchar(50)	YES				select,insert,update,references
vornname	varchar(50)	YES				select,insert,update,references
adresse	varchar(100)	YES				select,insert,update,references
email	varchar(50)	YES				select,insert,update,references
login	varchar(10)	YES				select,insert,update,references
password	varchar(10)	YES				select,insert,update,references

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
professor	0	PRIMARY	1	prof_id	A	1781				BTREE	

Database: wbs\_dokudb Table: rechnungen Rows: 3080

Field	Type	Null	Key	Default	Extra	Privileges					
id	int(11)										
k_id	int(10) unsigned		PRI	0	auto_increment	select,insert,update,references					
best_ids	varchar(200)					select,insert,update,references					
md	timestamp(14)		YES			select,insert,update,references					
rechdatum	date		YES			select,insert,update,references					
mehdatum	date		YES			select,insert,update,references					
status	enum('new','sent','done','mahn1','mahn2','storn')		YES			select,insert,update,references					
betrag	double(8,2)		YES			select,insert,update,references					
Table	Non unique	Key name	Seq_in_index	Column name	Collation	Cardinality	Sub_part	Packed	Null	Index type	Comment
rechnungen	0	PRIMARY	1	id	A	2878				BTREE	

Database: wbs\_dokudb Table: sammlungen Rows: 7

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned					
dok_id	int(10) unsigned				select,insert,update,references	
					select,insert,update,references	

Table has no keys

Database: wbs\_dokudb Table: students Rows: 13811

Field	Type	Null	Key	Default	Extra	Privileges
id	int(10) unsigned		PRI		auto_increment	select,insert,update,references
vorname	varchar(30)					select,insert,update,references
name	varchar(40)		MUL			select,insert,update,references
email	varchar(100)					select,insert,update,references
info	varchar(200)		YES			select,insert,update,references
sent_mails	int(10) unsigned		YES	0		select,insert,update,references
nospam	enum('false','true')		YES	false		select,insert,update,references

Table	Non unique	Key name	Seq_in_index	Column name	Collation	Cardinality	Sub_part	Packed	Null	Index type	Comment
students	0	PRIMARY	1	id	A	13754				BTREE	
students	0	name	1	name	A	13754				BTREE	
students	0	name	2	vorname	A	13754				BTREE	
students	0	name	3	email	A	13754				BTREE	

Database: wbs\_dokudb Table: studiengeange Rows: 27

Field	Type	Null	Key	Default	Extra	Privileges
abkuerz	varchar(10)					
studiengang	varchar(50)		PRI			select,insert,update,references
departement	varchar(10)	YES				select,insert,update,references

Table	Non unique	Key name	Seq_in_index	Column name	Collation	Cardinality	Sub_part	Packed	Null	Index type	Comment
studiengeange	0	PRIMARY	1	abkuerz	A	27				BTREE	

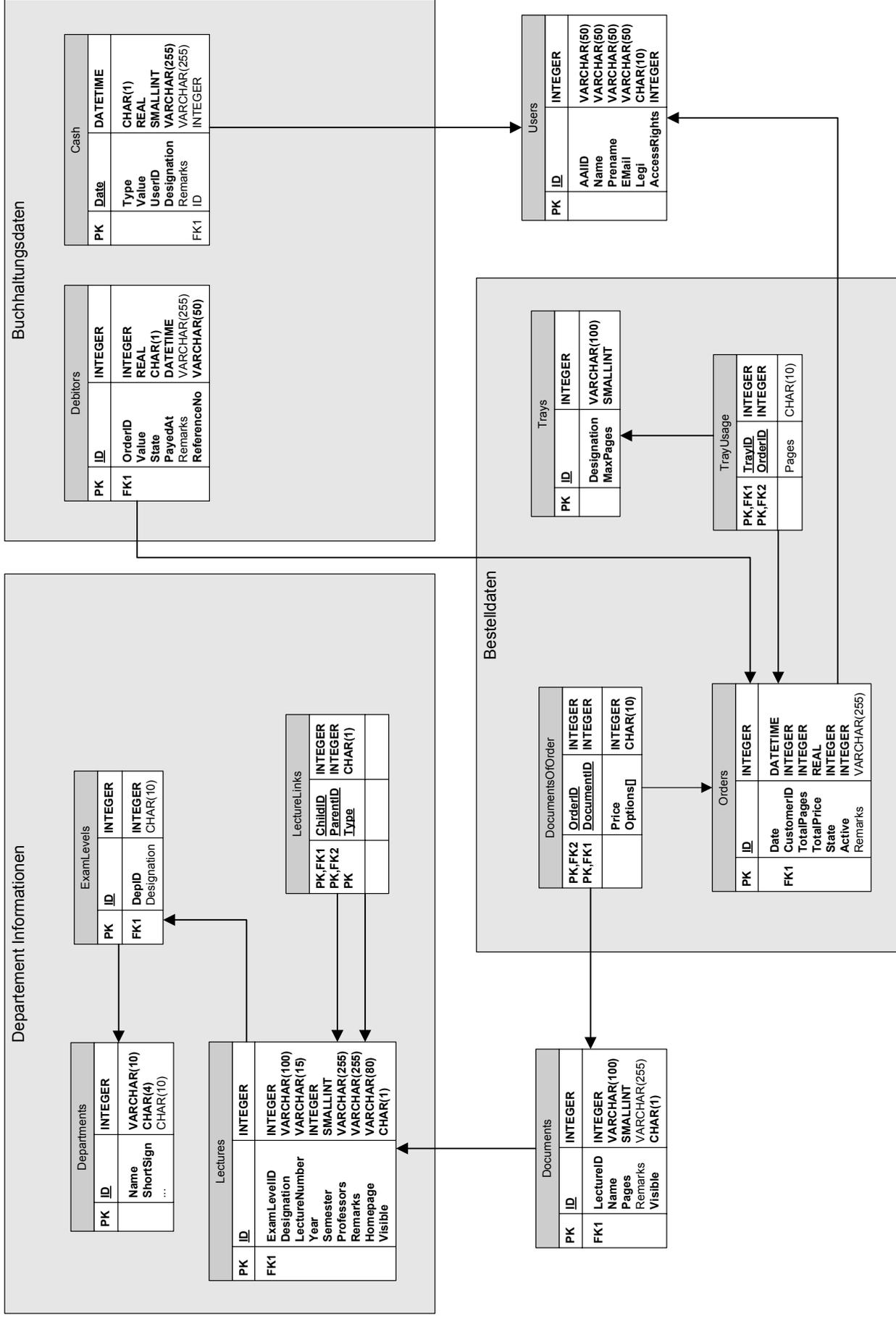
Database: wbs dokueb Table: vorlesung Rows: 1221

Field	Type	Null	Key	Default	Extra	Privileges				
v_nummer	varchar(25)		PRI			select,insert,update,references				
name	varchar(100)	YES				select,insert,update,references				
Table										
Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
0	PRIMARY	1	v_nummer	A	1221				BTREE	

Database: wbs dokueb Table: vorlesungsinstanz Rows: 1577

Field	Type	Null	Key	Default	Extra	Privileges				
vi_id	int(11)		PRI		auto_increment	select,insert,update,references				
md	timestamp(14)	YES				select,insert,update,references				
cd	timestamp(14)	YES		0000000000000000		select,insert,update,references				
v_nummer	varchar(25)	YES	MUL			select,insert,update,references				
jahr	varchar(9)					select,insert,update,references				
semester	char(2)					select,insert,update,references				
homepage	varchar(150)					select,insert,update,references				
beschreibung	text					select,insert,update,references				
studiengang	set('arch','baug','mavt','libet','infk','werk','bepr','math','phys','chem','biol','anbi','erdw','umnw','agrl','fowi','elek')					select,insert,update,references				
pruefungstufe	varchar(20)	YES				select,insert,update,references				
Table										
Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment
0	PRIMARY	1	vi_id	A	1577				BTREE	
0	v_nummer_2	1	v_nummer	A	788			YES	BTREE	
0	v_nummer_2	2	jahr	A	1577				BTREE	
0	v_nummer_2	3	semester	A	1577				BTREE	
0	v_nummer_2	4	studiengang	A	1577				BTREE	
1	v_nummer	1	v_nummer	A	788			YES	BTREE	

# B Neues Datenbank Schema



## C Referenzen

- 
- <sup>1</sup> Web Based Printing (WBS)  
<http://wbs.ethz.ch/>
  - <sup>2</sup> nethz Ein Portal für Dienstleistungen für Studierende. Erlaubt z.B. das Anmelden an diverse ETH-Dienstleistungen über einen globalen Login.  
<http://n.ethz.ch/>
  - <sup>3</sup> SPOD - Student Print On Demand  
<http://spod.ethz.ch/>
  - <sup>4</sup> GnuCash – Open Source Accounting Software  
<http://www.gnucash.org/>
  - <sup>5</sup> Phrame Application Framework Project  
<https://www.phrame.org/>
  - <sup>6</sup> Wordpress Blog  
<http://wordpress.com/>
  - <sup>7</sup> Authentication and Authorization Infrastructure (AAI)  
<http://www.switch.ch/aai/>
  - <sup>8</sup> Summary - Shibboleth Login Procedure  
[http://www.switch.ch/aai/demo/demo\\_intro.html](http://www.switch.ch/aai/demo/demo_intro.html)
  - <sup>9</sup> CUPS: Common UNIX Printing System  
<http://www.cups.org/>
  - <sup>10</sup> CUPS Software Programmers Manual  
[http://www.cups.org/doc-1.1/spm.html#4\\_1\\_6](http://www.cups.org/doc-1.1/spm.html#4_1_6)