

# **Reimplementierung eines Print Output System unter GNU/Linux**

Diplomarbeit für die Prüfung zum Diplom-Wirtschaftsinformatiker (BA)  
an der Berufsakademie Mosbach.

von Benjamin Schweizer

## Vorwort

Im Dezember 1995 hatte ich meinen ersten Kontakt mit freier Software. Mein damaliger Mitschüler Andreas Turtschan hatte von einem neuen Betriebssystem mit Namen Linux gelesen und motivierte mich dazu, dieses System näher anzuschauen. Zu dieser Zeit war Kernel 1.2 aktuell und an KDE oder Gnome wagte noch niemand zu denken. So verbrachten wir ganze zwei Tage mit einer Grundinstallation, ohne zu wissen was uns tatsächlich erwartete.

In mir begann eine Leidenschaft zu wachsen. Sie brachte mich zu meiner Ausbildungsfirma, der dsb AG, wo ich das Studium der Wirtschaftsinformatik begann. Dort liegt mein Zuständigkeitsbereich mittlerweile bei Netzwerken und Linux-Servern. Und noch immer ist es dieselbe Leidenschaft für Linux und freie Software, die mich antreibt.

Mit der Idee unser bisheriges Print Output System durch den Einsatz von Open Source Programmen abzulösen konnte mich mein Teamleiter Hans Platzer begeistern. Dabei handelte es sich gleichermaßen um eine interessante, wie auch herausfordernde Aufgabenstellung. Sie sollte mir die Möglichkeit geben, die im Studium und im Betrieb gelernten Kenntnisse an einem eigenen Projekt auf die Probe zu stellen.

Konfrontiert mit vielen Jahren Softwareevolution, galt es zunächst, die Zusammenhänge und Abhängigkeiten zwischen den einzelnen Komponenten zu verstehen. Die vielfältigen Entwicklungen auf diesem Gebiet machten es manchmal schwierig, den Überblick zu behalten. Um an verlässliche Informationen zu gelangen, war es oft notwendig die Quelltexte fremder Programme zu untersuchen. Einzelne Module traten zum Vorschein, die immer wieder ähnliche Ziele verfolgten und sich in ein Gesamtbild einreiheten. Sie galt es neu zu kombinieren und an die eigenen Bedürfnisse anzupassen, um so etwas Neues zu gestalten.

Bei all meinen Kollegen und Freunden, die mich während meiner Arbeit unterstützt haben, möchte ich mich an dieser Stelle herzlich bedanken. Mein ganz besonderer Dank gilt Joseph Andronic, der mit seiner konstruktiven Kritik versuchte, meinen Blick auf das Wesentliche zu lenken, und Kurt Pilchowski, der mir mit seinen Erfahrungen und Erwartungen stets beiseite stand.

Benjamin Schweizer

Neckarsulm, im September 2004.

## Inhaltsverzeichnis

1	Einleitung .....	1
1.1	Motivation.....	1
1.2	Zielsetzung und Gliederung.....	1
2	Grundlegendes .....	3
3	Situationsanalyse.....	5
3.1	Print Output bei der dsb AG .....	5
3.2	Anforderungen an ein modernes Print Output System.....	7
4	Druckausgabe unter Unix/Linux .....	9
4.1	Übersicht der Spoolerstandards .....	9
4.2	Spooling Systeme .....	10
4.2.1	Berkeley Spooler .....	10
4.2.2	System V Spooler.....	10
4.3	Filter und Druckersprachen .....	11
4.4	Druckerbeschreibungen .....	12
5	Vergleich verschiedener Komponenten.....	13
5.1	Einschränkungen .....	13
5.2	Spooler im Vergleich .....	13
5.2.1	CEPS – Cisco Enterprise Print System.....	14
5.2.2	CPS – Coherent Printing System.....	14
5.2.3	CUPS – Common Unix Printing System .....	14
5.2.4	GNUlpr – GNU Line Printer .....	15
5.2.5	LPD – Line Printer Daemon.....	15
5.2.6	LPRng – Next Generation Line Printer .....	16
5.2.7	PDQ – Print Don't Queue .....	16
5.2.8	PPR – PostScript Printer .....	17
5.3	Output Filter und Druckerbeschreibungen .....	18
5.3.1	CUPS (Filter und PPDs).....	18
5.3.2	foomatic (PPDs) .....	19
5.3.3	Gimp-Print (Filter und PPDs).....	19
5.3.4	ifhp (Filter).....	19
5.3.5	OMNI (Filter und PPDs).....	20
5.3.6	HPIJS (Filter und PPDs).....	20
5.4	Verwaltungssoftware .....	21
5.4.1	CUPS (web) .....	21
5.4.2	lpinfo .....	22

5.4.3	PPR-Web .....	23
5.5	Fazit .....	24
6	Eigenentwicklung auf Basis von CUPS .....	26
6.1	Technische Ausgangslage .....	26
6.1.1	Struktureller Aufbau .....	26
6.1.2	Softwareumgebung .....	26
6.2	Lizenzrechtliche Bestimmungen durch die GPL .....	27
6.2.1	Gleiches Recht für alle .....	27
6.2.2	Modifikation oder Eigenentwicklung .....	28
6.2.3	Veröffentlichung .....	28
6.3	Diskussion verschiedener Entwicklungsansätze .....	29
6.3.1	Schnittstellen und Aufsetzpunkte .....	29
6.3.2	Variante 1a: Customizing der Oberfläche .....	29
6.3.3	Variante 1b: Modifikation des Webservers .....	30
6.3.4	Variante 2a: Reimplementierung der Oberfläche .....	30
6.3.5	Variante 2b: Reimplementierung der Oberfläche und Kontrollprogramme .....	31
6.3.6	Fazit .....	32
6.4	Implementierung .....	32
6.4.1	Programmaufbau .....	32
6.4.2	Backend .....	34
6.4.3	Frontend .....	39
7	Resümee .....	44

## Glossar

3BSD	3 <sup>rd</sup> Berkeley Software Distribution
4BSD	4 <sup>th</sup> Berkeley Software Distribution
API	Application Programming Interface
ASP	Application Service Programming
CVS	Concurrent Versions System
GCC	Gnu C Compiler
ASCII	American Standard Code for Information Interchange
AFPL	Aladdin Free Public License
BSD	Berkeley Software Distribution
CAP	Columbia Appletalk Package
CEPS	Cisco Enterprise Print System
CGI	Common Gateway Interface
CPS	Coherent Print System
CICS	Customer Information Control System
CLI	Command Line Interceace
CUPS	Common Unix Printing System
EBCDIC	Extended Binary Coded Decimal Interchange Code
ESA	Enterprise System Architecture
FAQ	Frequently Asked Questions
GIMP	Gnu Image Manipulation Program
GNU	Gnu is Not Unix
GNUlpr	Gnu Line Printer
GPL	Gnu / General Public License
HPIJS	Hewlett-Packard InkJet Server
HTTP	HyperText Transport Protocol
IPP	Internet Printing Protocol
LP	Line Printer
LPD	Line Printer Daemon
LPRng	Line Printer Next Generation
LPR	Line Printer Remote
PDQ	Print Don't Queue
PHP	Php Hypertext Processor
POWER	Priority Output Writers, Execution processors, and input Readers
PPD	PostScript Printer Description
PPR	PostScript Printer

SNA	Systems Network Architecture
SOHO	Small Office and Home Office
SPOOL	Simultanous / Shared Peripheral Operation On-Line
SVR4	System V Release 4
URI	Unified Ressource Identificator
VM	Virtual Machine
VSE	Virtual Memory Extended
VTAM	Virtual Telecommunication Access Method

## Abbildungsverzeichnis

Abbildung 1: Gesamtübersicht .....	3
Abbildung 2: Aufbau des Host .....	5
Abbildung 3: VTAMPRINT - Queue Display .....	6
Abbildung 4: PRESCRIBE IIe Steuerkommandos.....	6
Abbildung 5: Filter.....	11
Abbildung 6: Output Filter .....	18
Abbildung 7: CUPS Weboberfläche .....	21
Abbildung 8: lpinfo .....	22
Abbildung 9: PPR-Web.....	23
Abbildung 10: Struktureller Aufbau .....	26
Abbildung 11: CUPS - Schnittstellen und Aufsetzpunkte .....	29
Abbildung 12: Aufbau Eigenentwicklung .....	31
Abbildung 13: Kommunikation mit CUPS - Variablendefinition.....	34
Abbildung 14: Kommunikation mit CUPS - IPP Anfrage.....	35
Abbildung 15: Kommunikation mit CUPS - Auswertung der Antwort .....	35
Abbildung 16: Kommunikation mit CUPS - Speicher aufräumen.....	36
Abbildung 17: Programmausgabe "getprinters demouser" .....	38
Abbildung 18: Aufbau des Frontends.....	39
Abbildung 19: Frontend - Dialogabfolge.....	41
Abbildung 20: Frontend - Druckerübersicht.....	42
Abbildung 21: Frontend - Auftragsübersicht.....	43

## **Tabellenverzeichnis**

Tabelle 1: Anforderungskatalog .....	8
Tabelle 2: Unices und deren Spooling Systeme.....	9
Tabelle 3: Printspooler - Schnittstellen.....	13
Tabelle 4: Printspooler - unterstützte Plattformen .....	17
Tabelle 5: Filtersammlungen.....	20
Tabelle 6: Frontend - Verarbeitungsfunktionen .....	40
Tabelle 7: Frontend - Eingabefunktionen .....	40

# **1 Einleitung**

## **1.1 Motivation**

Die dsb AG ist ein führender Application Service Provider für Medienunternehmen im Bereich Zeitschriftenverlage, Zeitschriftenhandel, Versandhandel und Abonnementmanagement. Wir bieten unseren Kunden unter anderem ein breites Spektrum an Dienstleistungen für den erfolgreichen Zeitschriftenvertrieb an.

Eine dieser Dienstleistungen ist die zentrale Verwaltung von Druckaufträgen über unser Print Output System. Bei der bisherigen Lösung setzen wir die Software „VTAMPRINT“ der Firma Macro4 ein. Sie läuft innerhalb einer VSE-Partition auf unserem IBM S/390 kompatiblen Comporex/Hitachi Mainframe.

Druckaufträge werden auf dem Mainframe während der Batchverarbeitung generiert und zur weiteren Bearbeitung an VTAMPRINT übergeben. Das Programm übernimmt die IP basierte Remote-Drucksteuerung verschiedener Netzwerkdrucker, die sowohl im Haus als auch an verschiedenen Standorten bei unseren Kunden in Europa stationiert sind. Weiter übernimmt „VTAMPRINT“ ein vollständiges mandantenfähiges Management der Printqueue. So ermöglicht es dem Kunden neben dem eigentlichen Drucken seiner Aufträge, diese im Printspooler zu priorisieren, einzusehen, abzurechnen und erneut zu drucken.

Im Zuge der unternehmensinternen Migration bisheriger Applikationen der bestehenden Mainframe-Umgebung auf eine IBM RS/6000 Serverplattform unter IBM AIX ist für den Bereich Print Output eine neue Lösung erforderlich. Neben neuen Funktionen und Schnittstellen ist es dabei wichtig, die bisherige Funktionalität beizubehalten und unseren Kunden dasselbe Maß an Verfügbarkeit, Performance und Qualität zu garantieren wie sie es bisher vom Großrechner gewohnt sind.

## **1.2 Zielsetzung und Gliederung**

Das bisher bei der dsb AG eingesetzte Print Output System „VTAMPRINT“ ist eine mainframebasierte Applikation, deren Aufgabe es ist, Druckaufträge aus Batchläufen zu verwalten und auf verschiedenen Druckern im Haus und beim Kunden auszugeben. Bedingt durch die Migration auf IBM AIX und die geplante Substitution des Großrechners muß auch das Print Output System überarbeitet werden.

Ziel dieser Arbeit ist es, die wesentlichen Merkmale der hostbasierten Lösung aufzuarbeiten und sowohl aktuelle technische Entwicklungen als auch veränderte Kundenanforderungen in die Neukonzeption einfließen zu lassen. Die Analyse des bisherigen Print Output Systems und das Design sowie die Implementierung einer neuen Applikation in einer GNU/Linux Umgebung sollen dabei im Mittelpunkt stehen.

Diese Arbeit ist folgendermaßen aufgebaut: Das erste Kapitel führt den Leser in die Thematik ein. Die Bedeutung des Print Outputs für die dsb AG wird aufgezeigt und es erfolgt eine Beschreibung der Aufgabenstellung.

Kapitel zwei beschäftigt sich mit der Druckausgabe im allgemeinen. Die wesentlichen Komponenten eines Print Output Systems werden anhand ihres geschichtlichen Kontextes vorgestellt. Es soll die Grundlagen vermitteln, die zum Verständnis der darauf folgenden Kapitel notwendig sind.

In Kapitel drei wird das bisher eingesetzte Print Output System VTAMPRINT betrachtet. Anhand von Erfahrungen aus der Praxis wird ein Anforderungskatalog definiert. Dieser gibt das Ziel für die Reimplementierung des Print Output Systems vor.

Das vierte Kapitel führt in die Druckausgabe unter Unix/Linux ein. Es werden die beiden wesentlichen Standards vorgestellt und ihre Verbreitung bei modernen Unices wird aufgezeigt. Weiter wird die Bedeutung von Filtern, Druckersprachen und Druckerbeschreibungen verdeutlicht.

Kapitel fünf vergleicht verschiedene Lösungsansätze. Dazu werden frei verfügbare Spooler, Filter und Druckerbeschreibungen gegenüber gestellt. Eventuelle Abhängigkeiten und Restriktionen werden untersucht und ihre Eignung für den Aufbau eines neuen Print Output Systems wird herausgearbeitet.

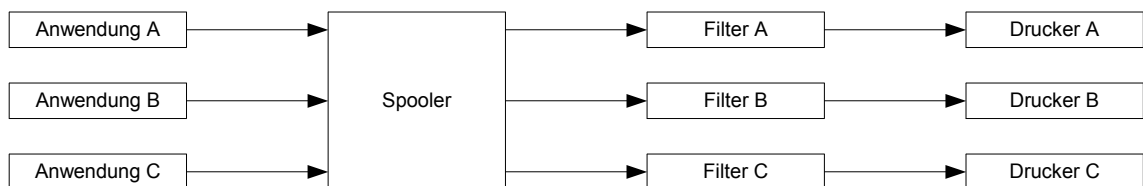
Das sechste Kapitel beschäftigt sich mit einer Eigententwicklung auf Basis von CUPS. Zunächst wird der technische Rahmen geklärt. Im Anschluß daran werden die rechtlichen Aspekte betrachtet und eventuelle Konflikte zwischen Open Source Lizenzen und kommerziellen Interessen aufgearbeitet. Es folgt eine Diskussion verschiedener Entwicklungsansätze und die Implementierung des vielversprechendsten.

Das siebte Kapitel faßt die gewonnenen Erkenntnisse zusammen. Das entwickelte Print Output System wird mit dem Anforderungskatalog verglichen.

## 2 Grundlegendes

In den Anfangsjahren der EDV waren sowohl Rechner als auch Drucker langsam. Die wenigen Anwendungen wurden direkt an die Hardware angepaßt, und ihren Output gaben sie sofort auf einem angeschlossenen Drucker aus. Dies hatte zwei entscheidende Nachteile: Zum einen war abzusehen, daß die Verbesserungen an der mechanische Ausgabe nicht mit dem technischen Fortschritt beim Rechnerbau mithalten konnten und sich hieraus ein Flaschenhals ergab; zum anderen konnte immer nur eine Anwendung auf einen Drucker zugreifen, die Druckernutzung war exklusiv.

Um Verarbeitung und Ausgabe von einander zu entkoppeln, wurde daher schon früh ein sogenannter „Spooler“ eingesetzt. Die Aufgabe des Spoolers ist es, Druckaufträge entgegenzunehmen, in einem Puffer zwischenspeichern und nacheinander an die Peripherie zu übergeben. Das *Handbook of IBM Terminology*<sup>1</sup> definiert Spool als „Simultaneous / Shared Peripheral Operation On-Line“; also der gleichzeitige oder geteilte Zugriff auf Ressourcen.



**Abbildung 1: Gesamtübersicht**

Neben dem Spooler stellen Filter eine wesentliche Komponente der Druckausgabe dar. Ihre Aufgabe ist es, die Druckdaten für einen bestimmten Drucker aufzuarbeiten und herstellerspezifische Steuersequenzen in den Datenstrom zu mischen. Früher waren diese Filter Teil der jeweiligen Anwendung. Um neue Hardware zu unterstützen, mußte jedes Programm einzeln angepaßt werden. Die zunehmende Druckervielfalt und der damit gestiegene Anpassungsaufwand führten dazu, daß diese Filter vermehrt an eine zentrale Stelle ausgelagert wurden. Angeordnet zwischen Spooler und Drucker können sie losgelöst von der Applikation ihre Aufgabe erledigen. Aus der Sichtweise einer Anwendung verhält sich so jeder Drucker gleich, und herstellerspezifische Steuersequenzen haben hier keine Bedeutung.

Aufbauend auf diesen Komponenten findet sich häufig noch weitere Verwaltungssoftware. Diese kann von System zu System sehr unterschiedlich aussehen und ist an die

---

<sup>1</sup> Siehe [Xep98]

jeweiligen Systemanforderungen angepaßt. Ihr Funktionsumfang reicht von einfachen Operationen, wie beispielsweise dem Repriorisieren oder Löschen von Aufträgen in der Warteschlange, bis hin zu komplexen Such- und Filterfunktionen wie sie bei VTAMPRINT zu finden sind.

### 3 Situationsanalyse

#### 3.1 Print Output bei der dsb AG

Die Druckausgabe bei der dsb AG ist bisher vollständig mainframebasiert. Auf dem im Haus eingesetzten S/390 kompatiblen Großrechner der Firma Compaq laufen mehrere VSE/ESA Instanzen unter dem Betriebssystem VM/ESA. Diese sind in mehrere Partitionen unterteilt, in denen neben den eigentlichen Anwendungen und Diensten auch der Spooler POWER (Priority Output Writers, Execution processors, and input Readers) und die Verwaltungssoftware VTAMPRINT laufen. Anwender können sich über das CICS (Customer Information Control System) mit VTAMPRINT verbinden und dort die komplette Verwaltung ihrer Druckaufträge vornehmen.

VM/ESA																																																		
VSE/ESA 1					VSE/ESA 2					VSE/ESA 3					VSE/ESA 4					VSE/ESA 5																														
P	O	W	E	R						V	T	A	M	P	R	I	N	T	N	A	T	U	R	A	L																									

**Abbildung 2: Aufbau des Host**

Druckaufträge werden in diesem Szenario während der Batchverarbeitung erzeugt, an den Spooler POWER übergeben und von diesem im Spool-Bereich bis zum endgültigen Druck vorgehalten. POWER unterscheidet dabei zwischen drei Datenbereichen: Queue File, Data File und Account File. Queue File und Account File enthalten Metainformationen wie Eigentümer, Erstelldatum, Größe und Status; im Data File liegen die eigentlichen Druckdaten.

Auf POWER aufsetzend arbeitet die Verwaltungssoftware VTAMPRINT. VTAMPRINT übernimmt die Kommunikation mit dem Anwender und erlaubt ihm vielfältige Operationen mit seinen Printjobs und Drucker durchzuführen. So kann er Aufträge zum Druck freigeben, anhalten, erneut drucken oder einen laufenden Druck abbrechen. Aufträge können auf andere Drucker umgelenkt werden, und über diverse Einstellungen ist es möglich, Seitenvorschub, Segmentierungsgröße, Anzahl der Kopien und die Priorität eines Auftrages einzustellen.

Neben diesen Operationen bietet VTAMPRINT dem Anwender die Möglichkeit, sich vor dem Druck einzelne Aufträge anzuschauen und die relevanten daraus zu selektieren. Verschiedene Ansichten und Suchfunktionen unterstützen ihn dabei, den Überblick über seine Daten zu behalten.

```
VTAMPRINT 4520G                LST Queue Display                11/08/2004 16:10:26
User: IPVX                      Terminal: BDSG30D0
  From User: *                   Subset: *                   Ext Writer: *
CMD   Jobname  Jobno P S Classes D Dest      Form Pages  Date   Time  Printer
  *   *       * * * * *         *
-----
IPLPN5T6 2340 5 N R          L R109      3751      0 06/08/04 20:36
IPVLIMT6 2341 5 N R          L R109      3751      8 06/08/04 20:37
IPV409T6 2342 5 N R          L R109      3751      1 06/08/04 20:46
IPV420T6 2343 5 N R          L R109      3751      1 06/08/04 20:47
IPVADLIX 2344 5 N R          L R109      3751      1 06/08/04 20:48
IPKVMHPR 2345 5 N R          L R109      3751      1 06/08/04 20:49
IPFEHLV6 2351 5 N R          L R109      3751      1 06/08/04 21:31
IPANLGV6 2352 5 N R          L R109      3751      3 06/08/04 21:34
IPANLGV6 2353 5 N R          L R109      3751      1 06/08/04 21:35
IPANLGV6 2354 5 N R          L R109      3751      1 06/08/04 21:35
IPANLGV6 2355 5 N R          L R109      3751      1 06/08/04 21:36
IPANLGV6 2356 5 N R          L R109      3751      1 06/08/04 21:37
IPANLGV6 2357 5 N R          L R109      3751      1 06/08/04 21:38
IPLPN5T9 2844 5 N R          L R109      3751      0 09/08/04 22:16

COMMAND ==>>
CMD: View(S) Uview Fview PPrint HOLD P URge eXtra retrieve(?) FULLON/OFF SHIFT
Pfk1      2          3 QUIT      4          5          6
Pfk7 UP    8 DOWN     9          10 VIEW    11         12
```

Abbildung 3: VTAMPRINT - Queue Display

Druckertreiber werden im bisherigen System ebenfalls in VTAMPRINT definiert. Bei der Ausgabe von Listen als ASCII-Text genügt es dazu im Regelfall einige Steuercodes vor den eigentlichen Daten an den Drucker zu senden. Dies ist exemplarisch in Abbildung 4 dargestellt. Es wird zunächst mit CSET eine deutsche Sonderzeichentabelle geladen. Anschließend lädt FSET eine Schrift fester Breite und EXIT beendet diese Sequenz.

```
!R! CSET 0G; FSET 0p10h12v0s3b3T; EXIT;
```

Abbildung 4: PRESCRIBE IIe Steuerkommandos<sup>2</sup>

<sup>2</sup> Siehe [Kyo98], PRESCRIBE IIe Kommandos.

### 3.2 Anforderungen an ein modernes Print Output System

Ein neues und modernes Print Output System muß sich stets an seinem Vorgänger messen. Neue Erkenntnisse und Technologien, veränderte Kundenanforderungen und frische Ideen beeinflussen den Anforderungskatalog ebenso wie technische Rahmenbedingungen und interne Gepflogenheiten. Bei der Festlegung des Anforderungskataloges ist es wichtig, die „richtige Mischung“ aus Notwendigem und Möglichem zu erzielen und dennoch genügend Freiraum für zukünftige Entwicklungen zu lassen.

Als Grundlage für die Definition des neuen Anforderungskataloges dienen die Leistungsmerkmale des Altsystems. Überholte Funktionalität wie die Unterstützung des SNA Protokolls kann zugunsten neuer Features aus der Liste gestrichen werden. Durch Gespräche mit den Betroffenen, den Anwendern und Administratoren, konnte auf ihre Erfahrungen und Wünsche eingegangen werden. Es entstand nachfolgender Anforderungskatalog:

Schlagwort und Beschreibung	Gewichtung
Anhalten und Fortsetzen von Aufträgen Der Druck von Aufträgen soll angehalten und zu einem späteren Zeitpunkt fortgesetzt werden können.	unwichtig
Auftragsübersicht Eine übersichtliche Darstellung der Druckaufträge soll dem Anwender helfen, einen Überblick seiner Druckaufträge zu behalten. Dazu werden Name, Erstell- und Druckdatum eines Auftrages angezeigt.	normal
Mandantenfähigkeit Für einen Application Service Provider (ASP) ist es zwingend notwendig konkurrierende Mandanten gegeneinander abzuschirmen. Auch dürfen sie nicht durch die Arbeit anderer Mandanten beeinträchtigt werden.	wichtig
Mehrsprachigkeit Die Bedienoberfläche muß für unsere Kunden in den Sprachen Deutsch, Englisch und Französisch verfügbar sein. Optional sollen weitere Sprachen hinzugefügt werden können.	wichtig
Priorisieren von Aufträgen Über die Vergabe von Prioritäten sollen Druckaufträge in der Warteschlange bevorzugt bearbeitet werden.	normal
Segmentierung von Druckaufträgen Das bisherige Print Output System kann Druckaufträge in Pakete zu je n Seiten zerlegen, wodurch sie einfacher in der Handhabung sind.	unwichtig

Seitenvorschau Der Anwender soll Aufträge im Spooler einsehen können und so bereits vor dem Druck entscheiden, ob es sich um den richtigen Druckauftrag handelt.	normal
Stichwortsuche Über eine Stichwortsuche innerhalb der Printqueue sollen Aufträge leicht auffindbar sein.	normal
Treiberunterstützung Alle im Einsatz befindlichen Drucker müssen vom Print Output System unterstützt werden.	wichtig
Umleiten von Aufträgen Es soll möglich sein, den Zieldrucker eines Druckauftrages zu ändern und diesen über einen anderen Drucker auszugeben.	normal
Wiederaufsetzen Nach einem Ausgabefehler soll es möglich sein, die Druckausgabe ab einer bestimmten Seite fortzusetzen.	mittel
Zuverlässigkeit Die Zuverlässigkeit des Print Output Systems steht mit an oberster Stelle.	wichtig

**Tabelle 1: Anforderungskatalog**

Neben den explizit im Anforderungskatalog genannten Punkten wurden bereits im Vorfeld einige Grundlagen geschaffen. Als Betriebssystemplattform wurde GNU/Linux festgelegt. Für die Realisation werden primär Open Source Technologien eingesetzt.

## 4 Druckausgabe unter Unix/Linux

### 4.1 Übersicht der Spoolerstandards

Das ursprünglich von Dennis Ritchie und Ken Thompson an den Bell-Labs entwickelte Betriebssystem Unix hat bis heute eine sehr wechselhafte Geschichte<sup>3</sup> durchlebt. Bedingt durch den ehemals frei verfügbaren Quelltext konnten Kopien des originalen Unix Time Sharing Systems vielerorts weiterentwickelt werden. Es entstanden viele unterschiedliche Zweige im Unix-Stammbaum, von denen die an der University of Berkeley, California entwickelte Berkeley Software Distribution (BSD) und das Unix System V (System V) die beiden bedeutendsten Vertreter darstellen.

Alle heute gebräuchlichen Unices basieren entweder direkt auf einem dieser beiden Zweige oder sie haben Programme und Schnittstellen daraus übernommen. An vielen Stellen, besonders aber im Bereich der Druckausgabe, wird daher zwischen Berkeley (BSD) und System V Release 4 (SVR4) Kompatibilität unterschieden. Die nachfolgende Tabelle zeigt auf, welche Spooler einige populäre Unices standardmäßig einsetzen.

Betriebssystem	Spooling System (Standard)
AIX (IBM)	System V, Berkeley
FreeBSD	Berkeley
GNU/Linux <sup>4</sup>	Berkeley
HP-UX (Hewlett Packard)	System V
IRIX (Silicon Graphics)	System V, Berkeley
NetBSD	Berkeley
OpenBSD	Berkeley
Solaris (Sun Microsystems)	System V
Tru64 (Hewlett Packard)	System V, Berkeley

**Tabelle 2: Unices und deren Spooling Systeme**

Neben den in Tabelle 2 aufgeführten Unices gibt es den freien Unix-Clone GNU/Linux. Linux ist dabei streng genommen nur der Kernel. Alle anderen Systemprogramme und die komplette Entwicklungsumgebung entstammen dem freien Betriebssystem GNU, das seit 1984 von der Free Software Foundation (FSF) entwickelt wird. Wegen des größtenteils frei verfügbaren Quelltextes von BSD konnten viele Programme für das GNU Betriebssystem einfach übernommen oder nachgeschrieben werden. Es entstand

---

<sup>3</sup> Siehe [Lev04], Unix Timeline

<sup>4</sup> Bei Verwendung des klassischen Berkeley Spooling System

eine enge Verwandtschaft zwischen beiden Systemen. Lange Zeit wurde daher der BSD Spooler als Standard-Spooler unter GNU/Linux eingesetzt.

## **4.2 Spooling Systeme**

### **4.2.1 Berkeley Spooler**

Mit Erscheinen der 3. Berkeley Software Distribution (3BSD) im März 1980 wurde das Berkeley Spooling System veröffentlicht. Es zeichnet sich durch seinen netzwerkzentrierten Ansatz aus und bildet heute den de-facto Standard in heterogenen Netzen. Neben dem originalen BSD Spooler lpd gibt es viele Systeme, die hierzu kompatibel sind.

Der BSD Print Spooler besteht aus zwei wesentlichen Programmen: der Serverkomponente lpd (Line Printer Daemon) und dem Client lpr (Line Printer Remote). Beide kommunizieren über das Line Printer Daemon Protocol<sup>5</sup>, das auch in nahezu allen Printservern und Netzwerkdruckern implementiert ist.

Hinzu kommen einige Verwaltungsprogramme, mit denen Druckaufträge aus der Warteschlange gelöscht oder angehalten werden können. Die Annahme neuer Druckaufträge kann verweigert und Drucker können angehalten werden. Sie bilden auf der Seite des Betriebssystems eine Schnittstelle hin zu den Anwendungen, die im Nachfolgenden als „Berkeley Interface“ bezeichnet wird.

Es gibt inzwischen neben dem original Berkeley Spooling System eine Reihe weiterer hierzu kompatibler Spooler. Sie werden in Kapitel 5.2 „Spooler im Vergleich“ näher untersucht.

### **4.2.2 System V Spooler**

Das in System V Release 4 (SVR4) enthaltene Spooling System lp stellt den zweiten bedeutenden Standard bei der Unix-Druckausgabe dar. Es wurde im Jahr 1986 veröffentlicht und unterscheidet sich bereits syntaktisch von seinem BSD Rivalen. Die Programmschnittstelle des System V Spoolers ist dadurch inkompatibel mit dem Berkeley Interface.

Der SVR4 Spooler gilt als leistungsfähiger im Vergleich mit dem BSD Spooler, aber auch als anspruchsvoller bei der Konfiguration. Der wesentlichste und auch auffälligste Unterschied besteht in der Anordnung von Druckern in Klassen. Durch sie ist es mög-

---

<sup>5</sup> Siehe [Lau90]

lich, mehrere Drucker in einer Gruppe zusammenzufassen. Bei Ausfall eines Druckers werden die Aufträge auf die anderen Maschinen der selber Klasse weitergereicht. Es entsteht eine Art von „Druck-Cluster“.

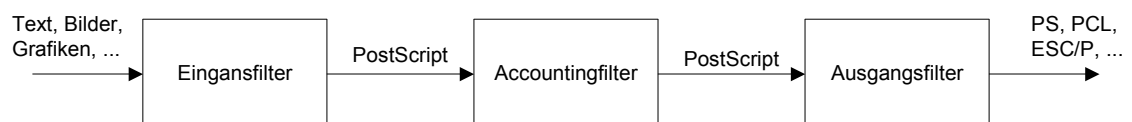
Auch für den SVR4 Spooler gibt es (freie) Alternativen und weiterentwickelte Versionen. Sie unterscheiden sich vor allen Dingen bei der Netzwerkanbindung. Das ursprüngliche System V Spooling System sieht keine direkten Netzwerkanbindungen vor. Soll ein dezentrales System in den Spooler integriert werden, muß dazu auf den Berkeley Spooler zurückgegriffen werden, und es entstehen hybride Installationen, die sich als sehr wartungsunfreundlich zeigen.

### 4.3 Filter und Druckersprachen

Ein Filter hat die Aufgabe, Druckdaten von einem Format in ein anderes zu überführen. Zu Zeiten, in denen lediglich ASCII-Text gedruckt wurde, genügte es durch entsprechende Steuersequenzen Schriftgröße, Ausrichtung und Seitenvorschub einzustellen. Mit der Anforderung, neben Text auch Grafik auszugeben, wurde dies jedoch komplizierter.

Für die Darstellung von Grafik ist es erforderlich, einen Drucker im Grafikmodus anzusprechen. Dies erfolgt über eine Druckersprache, die von Hersteller zu Hersteller unterschiedlich ist. Im professionellen Umfeld hat sich hierzu die Sprache Adobe PostScript<sup>6</sup> etabliert; im attraktiven SOHO-Segment<sup>7</sup> sind vermehrt Drucker mit Hewlett-Packards PCL und Epons ESC/P Druckersprache anzutreffen.

Je nach Installation und verwendetem Spooling System werden verschieden komplexe Filterstufen eingesetzt. Im Idealfall erzeugt eine Anwendung ihre Druckdaten bereits im PostScript Format. So können sie dann komplett ungefiltert zu einem PostScript-fähigen Drucker durchgereicht werden, und eine Filterung wird überflüssig. Man bezeichnet die hierbei verwendete Warteschlange auch als „RAW-Queue“.



**Abbildung 5: Filter**

---

<sup>6</sup> Siehe [Ado99]

<sup>7</sup> Small Office / Home Office

Ist ein einfaches Durchreichen der Druckdaten nicht möglich, müssen Filter zur Konvertierung in das entsprechende Zielformat eingesetzt werden. Damit die Anzahl der möglichen Kombinationen aus Quell- und Zielformat nicht über Hand nimmt, wird von den Filtern PostScript als Übergabeformat verwendet.

Es ist die Aufgabe des Eingangsfilters, aus beliebigen Eingangsdaten PostScript-Daten für die weitere Verarbeitung zu generieren. Die Aufgabe des Ausgangsfilters ist es, die PostScript Daten in die native Druckersprache zu übersetzen. Diese Funktionalität bezeichnet man auch als Rastern, da in dieser Stufe aus Vektorgrafiken Punktgrafiken erzeugt werden. Von PostScript Druckern wird dieser Schritt intern über eine sogenannte RIP Einheit (Raster Image Processor) vorgenommen. Bei nicht PostScript fähigen Druckern ist es Aufgabe des Ausgangsfilters, die nötige Interpretation vorzunehmen. Zwischen Ein- und Ausgangsfiler können noch weitere Filter geschaltet werden. Mit ihnen ist es möglich, einen Druckauftrag auf bestimmte Seitenbereiche einzuschränken oder die Seitenzahl für eine genaue Abrechnung zu erfassen.

### **4.4 Druckerbeschreibungen**

Jede Druckerfamilie verfügt über charakteristische Merkmale. Sie unterscheiden sich hinsichtlich des verwendeten Papierformats, der Seitengröße und Auflösung, den installierten Schriftarten, dem verwendeten Zeichensatz, dem Druckverfahren und der Papierzufuhr. Einige dieser Informationen werden von den verschiedenen Stufen der Druckausgabe benötigt und müssen zwingend angegeben werden. Andere sind rein optional und werden nur in Sonderfällen verwendet.

Um diese Informationen in einem einheitlichen Format zu repräsentieren und für eine Vielzahl von Anwendungen zugänglich zu machen, wurde von der Firma Adobe das PPD-Format (PostScript Printer Description) entwickelt. Es fand zunächst seine Verbreitung auf Apple- und Windows-Rechnern. Mit der Präsenz vieler PPD-Dateien wurde es aber schließlich auch von Unix Systemen adaptiert, und so wird es inzwischen von vielen Applikationen und Spoolern verwendet.

## 5 Vergleich verschiedener Komponenten

### 5.1 Einschränkungen

Alle im nachfolgenden betrachteten Programme und Projekte sind „frei“ im Sinne der Open Source Definition<sup>8</sup> der Open Source Initiative. Sie unterliegen der GNU General Public License (GPL), der BSD-Lizenz oder sind unter vergleichbaren Bedingungen verfügbar.

In den beiden vorangegangenen Kapiteln „Situationsanalyse“ und „Druckausgabe unter Unix/Linux“ wurden Zielsetzung und technischer Rahmen bereits definiert. Dieses Kapitel beschäftigt sich mit der Auswahl geeigneter Hilfsmittel. Die wesentlichen Komponenten dabei sind Spooler, Filter und Verwaltungssoftware. Sie lassen sich weitgehend frei miteinander kombinieren und werden separat verglichen.

### 5.2 Spooler im Vergleich

Name	Netzwerkschnittstelle					Programm- Interface	Lizenz
	LPR	IPP	App- Socket	Apple- talk	SMB / CIFS		
CEPS	ja	nein	ja	nein	nein	Berkeley	GPL
CPS	ja	nein	nein	nein	nein	Berkeley	frei
CUPS <sup>a</sup>	ja	ja	ja	ja <sup>b</sup>	ja	Berkeley, System V	GPL
GNUlpr <sup>c</sup>	ja	nein	nein	nein	nein	Berkeley	GPL
LPD	ja	nein	nein	nein	nein	Berkeley	BSD
LPRng <sup>a</sup>	ja	nein	ja	ja	nein	Berkeley, System V	GPL, Artistic License
PDQ <sup>c</sup>	ja	nein	nein	nein	nein	anderes	GPL
PPR	ja	nein <sup>d</sup>	ja	ja	ja	Berkeley, System V	BSD

**Tabelle 3: Printspooler - Schnittstellen**

- a: Neben der/den freien Versionen ist auch eine kommerzielle Variante verfügbar
- b: Der Hersteller empfiehlt, die Appletalk-Unterstützung nicht zu verwenden
- c: Das Projekt wird nicht weiterentwickelt
- d: Die Entwickler-Version im CVS enthält teilweise IPP Unterstützung

---

<sup>8</sup> Siehe [OSI04]

### 5.2.1 CEPS – Cisco Enterprise Print System

Das Cisco Enterprise Print System<sup>9</sup> ist für den Einsatz in großen Drucklandschaften gedacht. Es wurde von Damian Iveigh bei Cisco Systems entwickelt und ist inzwischen unter der GPL Lizenz frei verfügbar.

Im Fokus der Entwicklung standen die Anbindung von Netzwerkdruckern in einer vertrauenswürdigen Umgebung. Eine Beschränkung der Zugriffsrechte ist daher ebenso wenig vorgesehen, wie die Verwendung geschützter Netzwerkprotokolle. So spricht CEPS das BSD-LPR Protokoll und AppSocket. Auf die Implementierung neuerer Standards wurde aber bisher verzichtet. Die Entwicklung um CEPS scheint seit Sommer 2003 eingeschlafen zu sein.

### 5.2.2 CPS – Coherent Printing System

CPS ist das Coherent Printing System<sup>10</sup> von Thomas Worthington. Es besteht im Wesentlichen aus einer Sammlung von Perl-Skripten, die das Interface des BSD Print Spoolers nachbauen und seine Grundfunktionalität widerspiegeln. Es ist ausschließlich für den Einsatz in kleinen Umgebungen gedacht und nicht für den Betrieb eines professionellen Print Output Systems geeignet. Der Autor plant nach eigenen Angaben auch keine weiteren Funktionen hinzuzufügen.

### 5.2.3 CUPS – Common Unix Printing System

Das Common Unix Printing System<sup>11</sup> ist ein unter der GNU General Public License verfügbares Spooling System, das in den letzten Jahren zunehmende Beliebtheit erfahren hat. Es wird von Easy Software Products (ESP) entwickelt und ist neben der GPL-lizenzierten Version auch als kommerzielles Produkt verfügbar. Die freie Version hat bei vielen Desktop Linux Distributionen inzwischen den Line Printer Daemon als Standard-Spooler abgelöst.

CUPS zeichnet sich durch seine gute Implementierung des Internet Printing Protocol (IPP) aus, unterstützt darüber hinaus aber auch LPR, AppSocket, Appletalk und Windows-Druckfreigaben (SMB / CIFS). Bedingt durch die IPP Unterstützung ist es möglich, CUPS über eine Web-Oberfläche zu bedienen. Dies wird in Kapitel 5.4 betrachtet.

---

<sup>9</sup> Siehe [Ive04]

<sup>10</sup> Siehe [Wor03]

Das Programm verfügt über eine sehr gute Unterstützung von Druckerbeschreibungen im PPD Format. Einige PPD Dateien gehören bereits zum Lieferumfang, und durch die Kombination mit anderen Paketen ist es so möglich, eine Vielzahl von Druckern auf einfache Art und Weise einzubinden.

Intern verwendet CUPS genau wie der System V Spooler Druckerklassen. CUPS kann dementsprechend auch als System V Spooler angesprochen werden. Weiter stellt CUPS Kompatibilitätsprogramme bereit, über die der Spooler auch als Berkeley Spooler fungieren kann. Durch seine Vielseitigkeit ist CUPS einer der leistungsfähigsten Spooler.

### 5.2.4 GNUlpr – GNU Line Printer

Der GNU Line Printer<sup>12</sup> entstammt einem Projekt von Hewlett-Packard. Ziel war es, den alten BSD Print Spooler abzulösen. Im Projekt wurde die grundlegende Funktionalität des Berkeley Spoolers nachgebaut. Nach Erscheinen der Version 1.01 im November 2001 wurde die Weiterentwicklung plötzlich eingestellt.

### 5.2.5 LPD – Line Printer Daemon

Der ursprüngliche BSD Printspooler LPD<sup>13</sup> zeichnet sich vor allen Dingen durch seine weite Verbreitung und die liberalen Lizenzbedingungen aus. Sie ermöglichen es, den Quelltext auch ohne Weitergabe in eigene Produkte zu integrieren, vorausgesetzt, es wird auf die Entwickler hingewiesen.

Der Line Printer Daemon gilt inzwischen als technisch überholt und im Vergleich mit neueren Entwicklungen auch als komplizierter in der Bedienung. Ihm fehlt die Unterstützung neuer Netzwerkprotokolle ebenso wie die Integration von PPD Druckerbeschreibungen. Um neue Drucker einzubinden ist es deshalb oft erforderlich, eigene Filter-Skripte zu schreiben und ihn manuell zu konfigurieren.

Dank seiner früheren Beliebtheit ist der LPD jedoch ausgezeichnet dokumentiert, und es existieren für viele Problemstellungen bereits vorgefertigte Lösungen. Insgesamt setzt die Administration des Berkeley Spoolers jedoch ein hohes Know-How voraus.

---

<sup>11</sup> Siehe [Esp04]

<sup>12</sup> Siehe [Woo01]

<sup>13</sup> Siehe [Maj03]

Der LPD ist noch immer auf vielen älteren Systemen anzutreffen. Für die Realisierung eines neuen Print Output Systems ist er jedoch nicht die erste Wahl, da Spooler wie LPRng denselben Funktionsumfang abdecken und sicherer sind.

### 5.2.6 LPRng – Next Generation Line Printer

Patrick Powells LPRng<sup>14</sup> ist eine Reimplementierung des BSD Printspoolers. Er wurde von Grund auf neu geschrieben und dabei wurden einige Schwächen im Design ausgebessert. Viele Operationen, die beim Original Line Printer Daemon root-Rechte benötigen, können beim LPRng mit normalen Benutzerrechten durchgeführt werden.

Mit dem Redesign wurden auch die Unterstützung für Pretty Good Privacy (PGP) und Kerberos implementiert. Dies ermöglicht die Authentizität eines Benutzers sicherzustellen, wenn LPRng auf beiden Seiten eingesetzt wird.

Der Next Generation Line Printer ist in zwei freien Versionen verfügbar. Dem Anwender ist es überlassen, ob er die General Public License oder die Artistic License einsetzen möchte. Zusätzlich besteht die Möglichkeit, den Printspooler von AStArt Technologies kommerziell zu lizenzieren.

Insgesamt macht dieser Spooler einen sehr guten Eindruck. Er ist dem LPD vom Sicherheitskonzept her überlegen, verfügt jedoch über weniger Funktionen und Schnittstellen als CUPS oder PPR.

### 5.2.7 PDQ – Print Don't Queue

Der Printspooler PDQ<sup>15</sup> wird von seinem Entwickler Jacob A. Langford selbst als „Zielgerichtetes und hochflexibles Druckausgabesystem“<sup>16</sup> bezeichnet. Der Autor verwendet diese Aussage vermutlich mit einem Ironischen Unterton: Das Programm ist, wie bereits der Name andeutet, sehr schlank und verfügt über keinerlei Warteschlangen. Alle Druckaufträge werden direkt an den Drucker weitergeleitet. Im Singleuser Umfeld mag dies akzeptabel sein, ein professionelles Print Output System ist damit jedoch nicht realisierbar.

Die Entwicklung des GPL lizenzierten Programmes scheint inzwischen auch zum Erliegen gekommen zu sein. Die letzte stabile Version stammt vom März 2000.

---

<sup>14</sup> Siehe [Pow04]

<sup>15</sup> Siehe [Lan00]

### 5.2.8 PPR – PostScript Printer

Der PostScript Printer<sup>17</sup> von David Chappell wurde ursprünglich als Spooler für reine Postscript-Drucker entwickelt. In der aktuellen Version kann er über verschiedene Filter auch andere Modelle ansteuern. Seine Stärken liegen im Umgang mit fehlerhaften und beschädigten PostScript-Dateien. Diese werden gegebenenfalls isoliert und der Druck weiterer Aufträge wird unterdessen fortgesetzt. Administrative Eingriffe können so reduziert werden.

Die Unterstützung des PPD-Formats ermöglicht es, eine Vielzahl existierender Druckerbeschreibungen zu übernehmen und neue Drucker schnell in das System einzubinden. Genau wie CUPS enthält der PostScript Printer auch eine Web-Oberfläche. Sie wird in Kapitel 5.4 näher betrachtet.

Im Vergleich ist PPR neben LPRng und CUPS der leistungsfähigste Spooler. Bisher fehlt es ihm allerdings noch an der Unterstützung des Internet Printing Protocols. Im aktuellen CVS Snapshot<sup>18</sup> sind jedoch erste Schritte in diese Richtung verzeichnet.

Durch die Verfügbarkeit unter der BSD Lizenz ist dieser Spooler in manchen Fällen CUPS und LPRng vorzuziehen. Die BSD Lizenz verlangt im Gegensatz zur GPL keine Offenlegung des Quelltextes.

Name	AIX	HP-UX	Irix	Linux	Solaris	Tru64	xBSD
CEPS	- <sup>a</sup>	ja	- <sup>a</sup>	ja	ja	- <sup>a</sup>	ja
CPS	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	ja	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>
CUPS	ja	ja	ja	ja	ja	ja	ja
GNUlpr <sup>b</sup>	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>	ja	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>
LPD	ja	ja	ja	ja	ja	ja	ja
LPRng	ja	ja	ja	ja	ja	ja	ja
PDQ <sup>b</sup>	ja	ja	ja	ja	- <sup>a</sup>	- <sup>a</sup>	- <sup>a</sup>
PPR	- <sup>a</sup>	ja	ja	ja	ja	ja <sup>c</sup>	ja <sup>c</sup>

**Tabelle 4: Printspooler - unterstützte Plattformen**

- a: keine Auskunft
- b: Das Projekt wird nicht weiterentwickelt
- c: Es wurde eine frühere Version wurde getestet

---

<sup>16</sup> Siehe [Lan00], introduction.html

<sup>17</sup> Siehe [Cha04]

<sup>18</sup> CVS ist ein Versionsverwaltungssystem. Ein Snapshot ist eine Momentaufnahme daraus.

### 5.3 Output Filter und Druckerbeschreibungen

Ausgangs-Filter und Druckerbeschreibungen (PPDs) sind zwar grundsätzlich unabhängig von einander zu betrachten, werden aber in der Praxis häufig zusammen bereitgestellt. Dies liegt an der engen Verzahnung beider Komponenten. Erst ein speziell für einen Drucker erstelltes PPD oder ein Skript ermöglichen es, den entsprechenden Output Filter sinnvoll einzusetzen. Im allgemeinen werden für einen einzelnen Filter daher mehrere PPDs bereitgestellt, die das Zusammenspiel mit einem speziellen Druckermodell festlegen.

Im Gegensatz zum Spooler können unterschiedliche Filter- und PPD-Sammlungen gleichzeitig installiert werden. Filter präsentieren sich meist als Plugins für den PostScript Interpreter GhostScript. Sie ergänzen damit die bereits installierten Filter, verdrängen sie jedoch nicht. Für PPD Dateien genügt es, diese in einem speziellen Verzeichnis bereitzustellen. Gerade dieses Prinzip der Erweiterbarkeit mittels Plugins hat GhostScript zu einem de-facto Standard deklariert.

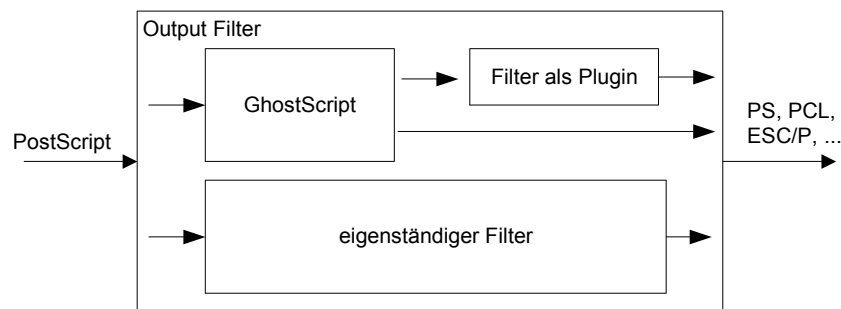


Abbildung 6: Output Filter

#### 5.3.1 CUPS (Filter und PPDs)

Das Common Unix Printing System bringt neben dem Spooler auch die Unterstützung für einige ausgewählte Druckermodelle mit. Neben PostScript Druckern sind Treiber für Dymo, Epson und HP Drucker im Standard-Paket enthalten. Der Hersteller ESP verwendet als Filtersoftware dazu eine speziell angepasste Version von Ghostscript, die unter dem Namen ESP Ghostscript mitgeliefert wird.

Die in CUPS enthaltenen Filter und PPDs genügen für viele Standard-Anwendungen, verzichten aber teilweise auf besondere Features. Erst die kommerzielle Version von CUPS, ESP Print Pro bringt weitere Druckerunterstützung mit.

### 5.3.2 foomatic (PPDs)

Till Kamppeter und Grand Taylor betreiben die Website „Linuxprinting.org“. Sie pflegen dort die „Printer Compatibility Database“, eine sehr umfangreiche Informationssammlung über verschiedene freie Filter und Druckermodelle. Aus dieser Datenbank können mit einigen Skripten PPD Dateien für die Druckeransteuerung generiert werden. Diese PPDs legen wiederum den zugehörigen Filter fest, den es separat zu installieren gilt. Als Filter werden Gimp-Print, Ghostscript und andere frei verfügbare Programme verwendet.

Über foomatic<sup>19</sup> ist es unter anderem auch möglich, Ghostscript Filter zu verwenden, die ohne eigene Druckerbeschreibungen ausgeliefert werden.

### 5.3.3 Gimp-Print (Filter und PPDs)

Das Gnu Image Manipulation Program (GIMP) hat mit Gimp-Print<sup>20</sup> seine eigenen Filter hervorgebracht. Sie gelten als sehr hochwertig und wurden mit dem Ziel entwickelt, eine professionelle Fotoausgabe unter Unix/Linux zu ermöglichen. Kurt Pfeifle vergleicht Gimp-Print im Handout des „CUPS and Foomatic“ Tutorials mit ihren Windows-Gegenstücken und bezeichnet sie als „gleichwertig oder sogar überlegen“<sup>21</sup>.

Die Gimp-Print Filter stellen eine Erweiterung für Ghostscript dar und können dadurch sehr einfach von vielen Spoolern verwendet werden. Durch gut 600 PPDs ist die Einbindung von Druckern der Firmen Canon, Epson, Lexmark, Sony, Olympus und Hewlett Packard elegant gelöst.

### 5.3.4 ifhp (Filter)

Ifhp<sup>22</sup> ist der Standard-Filter des LPRng Spoolers. Er ist gleichermaßen Eingang, Accounting und Ausgangsfilter und unterstützt die bidirektionale Kommunikation zwischen Spooler und Drucker. Für seine einzelnen Aufgaben greift ifhp auf verschiedene andere Programme zu, darunter auch auf den PostScript-Interpreter Ghostscript.

Das unter der General Public License und Artistic License stehende Programm dient zur Kapselung für verschiedene Systemprogramme. Es findet dort Einsatz, wo ein Spooler über keine native Unterstützung für den entsprechenden Filter verfügt.

---

<sup>19</sup> Siehe [Kam04]

<sup>20</sup> Siehe [Gim03]

<sup>21</sup> Siehe [Kpf02], Kapitel „Postscript and PPDs“, Abschnitt „Drivers and Printers in General“

<sup>22</sup> Siehe [Pow04]

### 5.3.5 OMNI (Filter und PPDs)

IBMs Linux Technology Center pflegt das Omni Printer Driver Model<sup>23</sup>. Die interessantesten Bestandteile von OMNI stellen in diesem Zusammenhang die 15 Filter und gut 470 Druckerbeschreibungen dar. Um die Druckerbeschreibungen verwenden zu können, muß man sie zunächst vom foomatic-kompatiblen XML Format in das PPD Format überführen. Anschließend können die Filter als GhostScript-Plugins angesprochen werden.

### 5.3.6 HPIJS (Filter und PPDs)

Das Hewlett-Packard Linux InkJet Project bietet mit dem InkJet Server<sup>24</sup> (HPIJS) Druckerunterstützung für eine Vielzahl von HP Druckern an. Neben Tintenstrahldruckern werden von dem Paket auch Laser- und Multifunktionsdrucker unterstützt. Insgesamt handelt es sich dabei um 291 Modelle.

Der HPIJS Treiber präsentiert sich als GhostScript-Plugin, welches intern verschiedene Filter-Modi für Laser-, Tintenstrahl und Multifunktionsdrucker unterscheidet. Alle Spooler, die GhostScript nutzen, können so von der HP Druckerunterstützung profitieren.

Paketname	Druckerbeschreibungen	Treiber (Filter)	Hersteller	Lizenz
foomatic	1315	- <sup>a</sup>	56	GPL
CUPS	13	3	4	GPL
Ghostscript	- <sup>b</sup>	27	-	GPL, AFPL
Gimp-Print	637	237	6	GPL
ifhp <sup>b</sup>	-	-	-	GPL, Artistic License
OMNI	472	15	10	LGPL
HPIJS	297	1 <sup>c</sup>	1 <sup>d</sup>	BSD

**Tabelle 5: Filtersammlungen**

- a: Filter müssen separat installiert werden und gehören nicht zu diesem Paket
- b: Dieses Paket verwendet als Output Filter Ghostscript
- c: Der Filter enthält verschiedene unabhängige Filter-Modi
- d: Offiziell werden nur Drucker von Hewlett-Packard unterstützt.

---

<sup>23</sup> Siehe [LTC03]

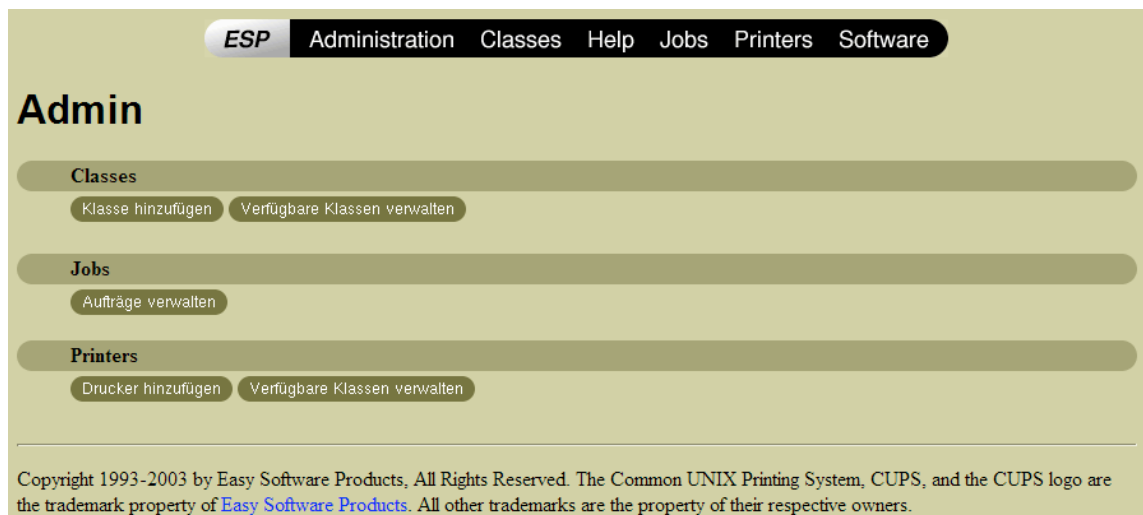
<sup>24</sup> Siehe [HPC04]

## 5.4 Verwaltungssoftware

Ein leistungsfähiges Print Output System zeichnet sich nicht zuletzt durch mächtige Werkzeuge aus, mit denen der Spooler und die Hilfsprogramme gesteuert werden können. Die kommandozeilenorientierten Utilities sind durch die Kompatibilität zum Berkeley Spooler oder System V Spooler bereits vorgegeben. Sie sind hervorragend zum Automatisieren von Aufgaben geeignet und stellen für Administratoren und anderen systemnahen Anwendern keine Hürde dar. Für Endkunden erscheint dieses Vorgehen jedoch recht umständlich. Sie erwarten eine zeitgemäße grafische Oberfläche, die keine große Einarbeitungszeit erfordert und mit der sie zielgerichtet ihre Arbeit erledigen können. Die Entwickler vieler Spooling Systeme haben dies erkannt und grafische Oberflächen, mit denen die Arbeit für Endanwender erleichtert wird, entwickelt.

Im Folgenden werden Verwaltungsprogramme untersucht, die über einen Webbrowser zu bedienen sind. Auf die Betrachtung rein Unix basierter Anwendungen soll hier bewusst verzichtet werden, da diese einen X Server voraussetzen, der einen nicht unerheblichen administrativen Aufwand und zusätzliche Kosten auf den meist Windows-basierten Clients nach sich ziehen würden.

### 5.4.1 CUPS (web)



**Abbildung 7: CUPS Weboberfläche**

Durch die Unterstützung des Internet Printing Protocols enthält CUPS einen integrierten Webserver. Über diesen ist es möglich, neue Drucker anzulegen und ihre Charakteristika zu definieren. Die Warteschlange kann gesperrt und angehalten werden, Auf-

träge können gelöscht und je nach Konfiguration auch erneut gedruckt werden. Für den Einsatz im kleinen ist diese Oberfläche eine gute Lösung.

Viele Funktionen, die in einer großen Installation benötigt werden, sind jedoch noch nicht integriert. So fehlen Funktionen wie eine Druckvorschau, das Umleiten von Aufträgen an andere Drucker, seitenweises Aufsetzen und eine saubere Abgrenzung einzelner Mandanten. Die CUPS Weboberfläche ist somit für Heimanwender gut geeignet, muß an Funktionalität aber noch einiges zulegen, um auch im professionellen Umfeld bestehen zu können.

### 5.4.2 Ipinfo

Alek Komarnitskys Ipinfo<sup>25</sup> ist eine Weboberfläche für die LPRng Spooling Software. Sie richtet sich sowohl an Administratoren, als auch an Anwender, die einen Überblick über ihre Druckaufträge benötigen. Ipinfo beobachtet dazu die Warteschlangen jedes einzelnen Druckers und kann die Anzahl der ausstehenden Druckaufträge anzeigen. Als Administrator können Druckaufträge gelöscht werden und über diverse Parameter kann die Druckerübersicht eingeschränkt werden.

The screenshot shows the Ipinfo web interface. At the top, there are navigation buttons: 'Help', 'Show', and a dropdown menu for 'printers in' set to 'ALL'. Below this, there are radio buttons for 'List only (fast!)', 'queue status' (which is selected), and 'active/broke'. A text input field for 'location' and 'or enter optional override substring:' is also present. Below the navigation is a 'Normal User' button and a link to 'Questions/suggestions on this web page can be directed to ccbox@hku.hk (2.40-)'. The main content area is titled 'ALL Printers (16 total)' and contains a table with the following data:

Queue Status	Printer Name	Description	Aliases
0 jobs	lp	No data available	Network printer Printronix P5203H
0 jobs	rr103bw1	Run Run Shaw rr103bw1 (HP 8000)	rr103bw1_1
0 jobs	rr103bw2	Run Run Shaw rr103bw2 (HP 8000)	rr103bw2_1
0 jobs	rr103bw3	Run Run Shaw rr103bw3 (HP 8000)	rr103bw3_1
0 jobs	olb134bw1	Old Library olb134bw1 (HP 8000)	olb134bw1_1
0 jobs	olb134bw2	Old Library olb134bw2 (HP 8000)	olb134bw2_1
0 jobs	olb134bw3	Old Library olb134bw3 (HP 8000)	olb134bw3_1
0 jobs	olb134bw3dp	No data available	olb134bw3dp_1

Below the table is a section titled 'lp Print Queue' with buttons for 'Check Queue every 60 seconds', 'More', 'Less', 'Queue Info', and 'Print TestPage'. The queue information is displayed as follows:

```
Printer: lp@hkusug (dest dlprn@colpsrv) 'Network printer Printronix P5203H'
Queue: no printable jobs in queue
Status: subsystem pid 6101 exit status 'JABORT' at 13:19:35.741
status: idle
interface: online
control: mode(default)
printer lcd: ONLINE
: POWER SAVER MODE
printer job: Done
```

Abbildung 8: Ipinfo<sup>26</sup>

<sup>25</sup> Siehe [Kom00]

Die Dokumentation zu lpinfo spricht von Installationen mit über 1000 installierten Warteschlangen, die mit Hilfe des Programmes verwaltet werden. In einem solchen Szenario zeigen sich jedoch schnell die Schwächen: Neben der fehlenden Mandantenfähigkeit und dem allgemein sehr geringen Funktionsumfang ist hier die schlechte Performance zu nennen; zum Aufbau der Statusanzeige müssen alle Druckerwarteschlangen angefragt werden. Dies erfolgt seriell und dauert ungefähr eine Sekunde je Gerät – die daraus resultierende Wartezeit macht die Arbeit in größeren Umgebungen ineffizient.

### 5.4.3 PPR-Web

PPR-Web ist die Web-basierte Oberfläche des PostScript Printers PPR. Als fester Bestandteil des PPR Paketes ist die Anwendung genau an dessen Schnittstellen angepaßt. Über einen eigenen Mini-Webserver „ppr-httpd“ werden verschiedene Perl-CGI Programme ausgeführt, die wiederum mit dem Spooler kommunizieren. Sie führen verschiedene Operationen durch und bereiten die Antworten vom Spooler grafisch auf.

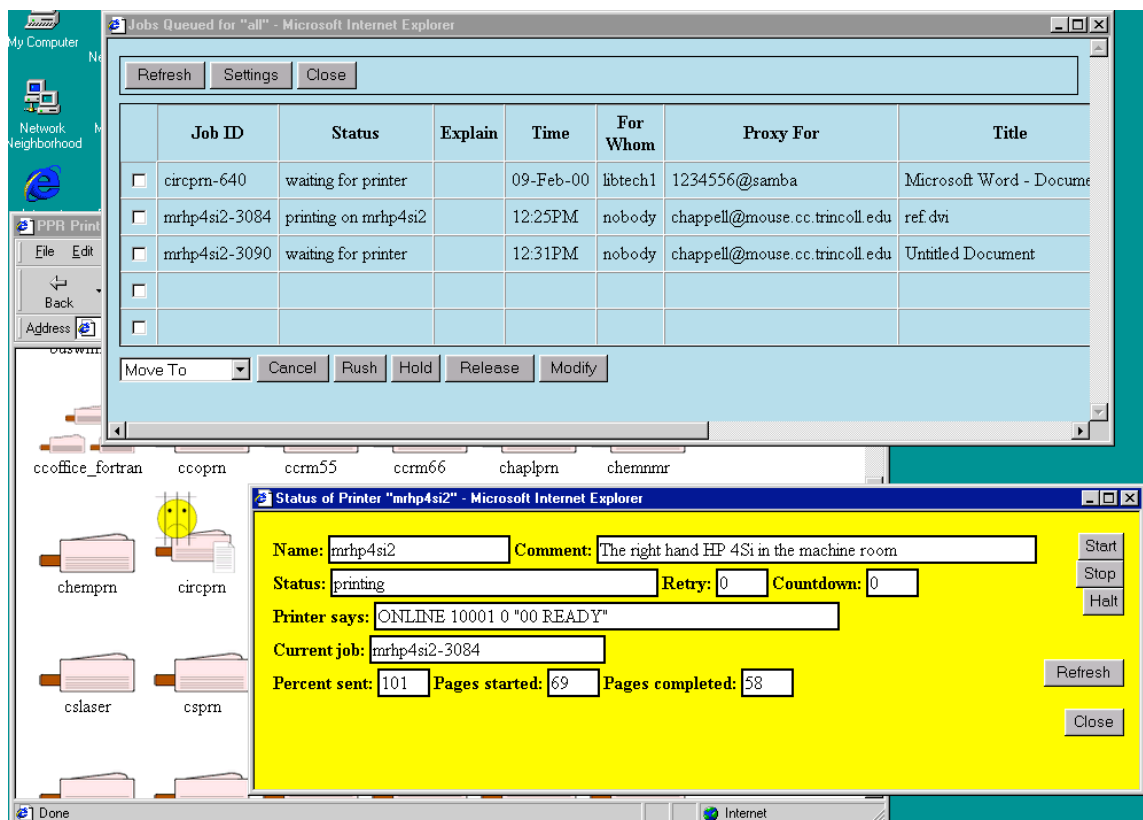


Abbildung 9: PPR-Web<sup>27</sup>

<sup>26</sup> Screenshot einer Installation an der University of Hongkong

<sup>27</sup> Screenshot entnommen aus [Cha04]

Nach erfolgreicher Authentifizierung können vom Administrator Drucker angezeigt, angehalten und wieder gestartet werden. Neue Drucker können über PPR-Web angelegt und alte gelöscht werden. Es ist möglich, die Warteschlange anzuzeigen und Druckaufträge zu löschen. Daneben gibt die Anwendung Auskunft über verschiedene Statusinformationen wie etwa den belegten Speicherplatz und die Logfiles.

Die Oberfläche richtet sich sowohl an Administratoren, die einen Überblick über alle im System vorhandenen Drucker benötigen, als auch an Anwender, die mit eingeschränkten Rechten ihre Aufträge bearbeiten können.

Genau wie bei der CUPS Oberfläche sind in PPR-Web viele nützliche Funktionen integriert, die es für einen Einsatz im kleinen qualifizieren. Einige erweiterte Funktionen wie die Druckvorschau und die Möglichkeit des Wiederaufsetzens abgebrochener Aufträge fehlen jedoch. Größtes Manko ist aber die fehlende Mandantenfähigkeit, also die strikte Unterscheidung verschiedener Benutzergruppen.

### **5.5 Fazit**

Von den untersuchten Print Spoolern machen CUPS, LPRng und PPR einen ausgezeichneten Eindruck. Sie erscheinen für den Aufbau eines neuen Print Output Systems am besten geeignet. Es handelt sich bei allen Spoolern um ausgereifte und in der Praxis erprobte Programme, die jeweils ihre individuellen Stärken haben.

CUPS und PPR heben sich von LPRng durch ihre breitere Protokollunterstützung ab, während LPRng recht klein ist und auf ohnehin oft unnötigen „Ballast“ wie beispielsweise einen eigenen Webserver verzichtet. PPR kann wegen der verwendeten BSD Lizenz sehr gut in kommerziellen Projekten verwendet werden, ist CUPS aber technisch unterlegen. Auf Grund der guten Integration von CUPS in einige neue Linux Distributionen und der sauberen Implementierung des Spoolers wird CUPS als Basis für ein neues Print Output System ausgewählt.

Durch die Entscheidung für das Common Unix Printing System ist auch die Schnittstelle zu Filtern und Druckerbeschreibungen vorgegeben. CUPS greift an dieser Stelle auf GhostScript zurück. So ist es möglich, alle zuvor vorgestellten Output Filter einzusetzen. Diese können parallel installiert werden. Abhängig von den angeschlossenen Druckermodellen muß lediglich der entsprechende Drucker konfiguriert werden.

Die Entscheidung für CUPS bedingt weiter auch eine Vorentscheidung für die Bedienoberfläche. Die untersuchten Web-Oberflächen lpinfo, PPR-Web und CUPS Web Interface sind sehr eng mit den zugrundeliegenden Spoolern verbunden; es ist nicht ohne

weiteres möglich, diese mit einem anderen Spooler einzusetzen. Speziell fortgeschrittene Funktionen wie seitenweises Aufsetzen, eine Druckvorschau im Spooler und allen voran die Trennung einzelner Mandanten werden von keiner Anwendung bereitgestellt.

Insgesamt ist CUPS jedoch eine solide Basis. Durch entsprechendes Customizing und Überarbeitung einiger Programmteile wird es an die Anforderungen und Bedürfnisse der dsb AG angepaßt. Zusammen mit einigen anderen Open Source Programmen wird es die Kernkomponente bilden, welche die einzelnen Bestandteile zusammenführt.

Besonderes Augenmerk liegt dabei auf einer funktionalen Oberfläche, über die unsere Kunden ihre Druckaufträge selbst verwalten können.

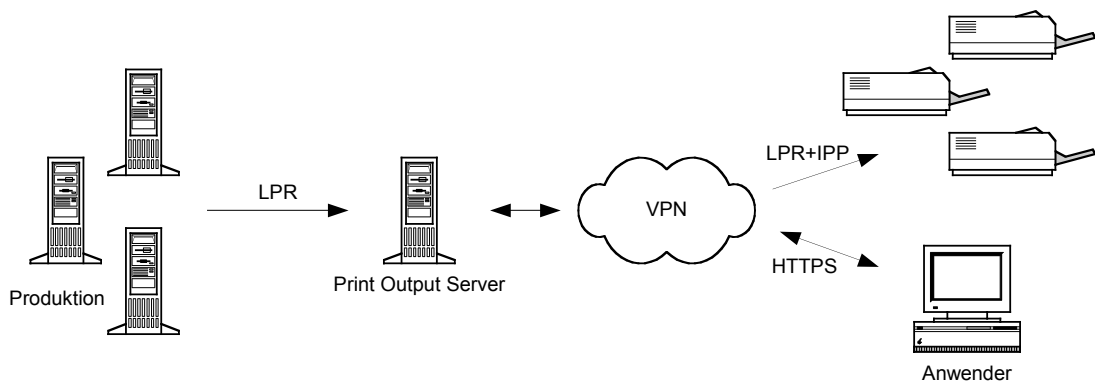
## 6 Eigenentwicklung auf Basis von CUPS

### 6.1 Technische Ausgangslage

#### 6.1.1 Struktureller Aufbau

Die Druckdaten werden zukünftig primär auf einer IBM RS/6000 Serverlandschaft erzeugt. Dort laufen die Anwendungen VHS/3 und ZHS/3 unter IBMs hauseigenem Unix AIX.

Die Druckdaten werden zunächst über das LPR Protokoll an den Print Output Server übergeben. Hier werden die Aufträge von den Anwendern selbst verwaltet. Dazu greifen diese über eine verschlüsselte Verbindung auf eine Web Application zu.



**Abbildung 10: Struktureller Aufbau**

Nachdem die Druckdaten von ihrem Ausgangsformat in die native Druckersprache umgewandelt sind, werden sie an die Drucker gespooled. Dazu wird innerhalb eines Virtual Private Network (VPN), das die dsb AG für ihre Kunden betreibt, das LPR Protokoll eingesetzt. Alternativ ist es auch möglich, das modernere Internet Printing Protocol einzusetzen, soweit es von den Druckern unterstützt wird.

#### 6.1.2 Softwareumgebung

Da CUPS vollständig in C geschrieben ist, wird eine C Entwicklungsumgebung bestehend aus Bibliotheken, Compiler und Hilfsprogrammen benötigt, um aus dem Quelltext ein lauffähiges Programm zu erzeugen. Durch die Wahl von GNU/Linux als Betriebssystemplattform bietet es sich an, an dieser Stelle die GNU Entwicklungsumgebung einzusetzen. Die im Hause dsb favorisierte Distribution SuSE Linux Standard Server 8 (SLSS8) enthält alle dafür notwendigen Programmpakete und -quellen.

Mit den im Standardumfang enthaltenen Filtern können bereits PostScript und PCL5 Drucker angesteuert werden. Da dies zum aktuellen Zeitpunkt für alle bei der dsb AG beziehungsweise beim Kunden im Einsatz befindlichen Drucker ausreichend ist, wird vorerst auf die Installation zusätzlicher Pakete verzichtet. Diese können gegebenenfalls zu einem späteren Zeitpunkt nachinstalliert werden.

## **6.2 Lizenzrechtliche Bestimmungen durch die GPL**

Das Common Unix Printing System steht unter der GNU General Public License<sup>28</sup> (GPL). Sie ist wohl die bekannteste Open Source Lizenz, der unter anderem auch Linux und GNU unterstehen. Die Verwendung von freier Software im kommerziellen Umfeld erfordert daher besondere Aufmerksamkeit und eine genaue Prüfung, wie sich Firmeninteressen und die Interessen der Entwickler vereinen lassen.

### 6.2.1 Gleiches Recht für alle

Die GPL besagt im wesentlichen, daß jeder das Recht hat, ein Programm zu verwenden, es zu verändern und beliebig weiterzugeben. Neben diesen Rechten werden mit Akzeptanz der GPL auch einige Pflichten angenommen. So muß zusammen mit dem Programm stets der zugehörige Quelltext weitergegeben werden, und der Empfänger genießt dieselben Rechte wie derjenige, der das Programm weitergibt. Dadurch soll die Freiheit dauerhaft gewährleistet werden, und es ist nicht möglich, freie Software in unfreie umzuwandeln.

Dies bedeutet auch, daß alle Änderungen am ursprünglichen Programm der GPL unterliegen und im Fall einer Weitergabe veröffentlicht werden müssen. Eine Pflicht zur Weitergabe besteht jedoch nicht. Wird eine modifizierte Version eines GPL lizenzierten Programmes nur intern eingesetzt, haben außen Stehende keinerlei Rechte daran. Dies ist beim Print Output System der dsb AG der Fall.

---

<sup>28</sup> Siehe [FSF91]

## 6.2.2 Modifikation oder Eigenentwicklung

Es stellt sich weiter die Frage, was eine Veränderung des ursprünglichen Programmes ist und was eine Eigenentwicklung davon abgrenzt. Die Frequently Asked Questions<sup>29</sup> (FAQ) zur GPL geben dazu Auskunft:

- Werden Programme gegen eine GPL lizenzierte Bibliothek gebunden, stellen sie eine abgeleitete Arbeit dar und unterliegen dadurch selbst den Bestimmungen der GPL
- Programme, die auf GPL Software zurückgreifen, beispielsweise über eine Kommandozeilenschnittstelle (CLI), sind als eigenständige Werke zu betrachten.
- Greifen Programme über eine Kommandozeilenschnittstelle direkt auf interne Datenstrukturen eines GPL Programmes zu, sind diese als kombinierte Werke zu betrachten und fallen damit unter die Bestimmungen der GPL.

## 6.2.3 Veröffentlichung

Unabhängig davon, ob die angestrebte Entwicklung von der GPL betroffen ist, muß geklärt werden, was einen internen Einsatz von einer Veröffentlichung unterscheidet. Bei einer Web Applikation erhält der Benutzer keinen Zugang zum Programm. Er bekommt lediglich die Programmausgabe präsentiert. Dies ist vergleichbar mit einem Brief, der über eine Textverarbeitung erzeugt wird oder ein Programm, das von einem Compiler übersetzt wird. Der Einsatz von GPL Software scheint somit für einen Application Service Provider (ASP) unbedenklich.

Die Free Software Foundation, Herausgeber der General Public License, scheint sich dessen bewußt zu sein. In der GPL FAQ wird explizit darauf eingegangen, dort steht<sup>30</sup>: "The GPL permits anyone to make a modified version and use it without ever distributing it to others."

---

<sup>29</sup> Siehe [FSF04]

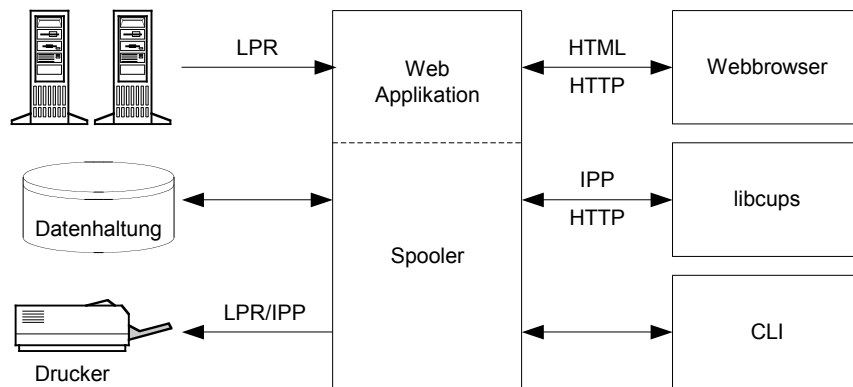
<sup>30</sup> Siehe [FSF04], Überschrift „Unreleased Mods“

## 6.3 Diskussion verschiedener Entwicklungsansätze

### 6.3.1 Schnittstellen und Aufsetzpunkte

Als zentrale Komponente steht der CUPS Spooler im Mittelpunkt der nachfolgenden Betrachtung. Er verbindet Drucker, Datenhaltung, Netzwerk und Anwender. Um die bisherige Codebasis an die gestellten Anforderungen anzupassen, ist es notwendig, neuen Programmcode in das gegebene Umfeld zu integrieren.

Eine Untersuchung des CUPS Quelltextes und der umfangreichen Dokumentation<sup>31</sup> zeigen, daß es verschiedene Stellen gibt, an denen auf den Spooler bzw. die bestehende Weboberfläche aufgesetzt werden kann.



**Abbildung 11: CUPS - Schnittstellen und Aufsetzpunkte**

Abbildung 11 veranschaulicht den strukturellen Aufbau. Über die integrierte Weboberfläche kann auf die wesentlichen Funktionen zugegriffen werden. Eine Kommandozeilenschnittstelle ermöglicht die flexible Bedienung fortgeschrittener Funktionen, die nicht über einen Browser erreicht werden können. Über das Internet Printing Protocol können annähernd alle Funktionen des Spoolers erreicht werden. CUPS enthält dazu eigens eine Programmbibliothek mit dem Namen „libcups“. Es ergeben sich die nachfolgend aufgezeigten Möglichkeiten.

### 6.3.2 Variante 1a: Customizing der Oberfläche

Die CUPS Weboberfläche basiert auf einem Template-System, das einfache HTML-Vorlagen zu vollständigen Webseiten zusammensetzt. Sie können angepaßt oder ausgetauscht werden, und so ist es möglich, jede Seite um neue Inhalte zu ergänzen.

<sup>31</sup> Siehe [Swe01] und [ESP03]

Die einfachste Lösung besteht darin, die vorhandenen Templates durch eigene zu ersetzen. Die zugrundeliegende Funktionalität wird dabei beibehalten. Alle noch fehlenden Funktionen werden separat implementiert und über externe Links in die bestehende Oberfläche integriert. Dazu ist eine eigenständige Web Applikation erforderlich, die diese zusätzlichen Funktionen bereitstellt.

Dieser Lösungsansatz zeichnet sich weniger durch technische Eleganz als viel mehr durch den geringen Entwicklungsaufwand aus. Die gewünschte Mandantenfähigkeit ist damit nur schwer zu realisieren; es müßten für jeden Mandanten eine eigene Instanz des Spoolers und des Webserver gestartet werden, und dies ist auf Performancegründen nicht wünschenswert.

Durch das „Springen“ zwischen zwei Applikationen entstehen nebenbei auch Probleme bei der Authentifizierung. Ohne doppelte Anmeldung ist es ohne weiteres nicht möglich, die Authentizität eines Anwenders festzustellen.

### 6.3.3 Variante 1b: Modifikation des Webserver

Eine Weiterentwicklung des Lösungsansatzes aus Variante 1a besteht darin, die fehlende Funktionalität nicht in einer eigenständigen Applikation unterzubringen, sondern den vorhandenen Webserver dementsprechend anzupassen. Es wird nur eine Kommunikationsschnittstelle verwendet, und so ist keine parallele Applikation notwendig.

Aufbauend auf der vorhandenen Codebasis kann alle Funktionalität direkt im CUPS Webserver untergebracht werden. Durch Zugriff auf die CUPS-internen Datenstrukturen erreicht man maximale Flexibilität. Die Mandantenfähigkeit, und Authentifizierung wird dadurch an einer zentralen Stelle abgehandelt.

Ein solches Vorgehen erscheint technisch elegant. Nachteilig daran ist der daraus resultierende hohe Wartungsaufwand. Eine Abspaltung vom ursprünglichen Entwicklungszweig hat zur Folge, daß zukünftige Veränderungen nicht einfach übernommen werden können. Sie müssen zwischen den beiden Zweigen angepaßt werden, und es besteht leicht die Gefahr, daß die Entwicklungen zu sehr auseinanderdriften.

### 6.3.4 Variante 2a: Reimplementierung der Oberfläche

Einen komplett anderen Ansatz stellt die Reimplementierung der Verwaltungssoftware dar. Durch eine Eigenentwicklung kann der Programmablauf optimal an unsere Bedürfnisse angepaßt werden, und die Entwicklung erfolgt losgelöst vom ursprünglichen

Quelltext. Die beiden separaten Programme teilen sich dann klar definierte Schnittstellen, können aber unabhängig voneinander weiterentwickelt werden.

Als Schnittstelle bietet sich das Kommandozeileninterface (CLI) an, das über eine Reihe von Systemprogrammen zugänglich ist. Diese entsprechen dem System V Spooler Interface und stellen damit einen quasi Standard dar, der relativ zukunftssicher ist. Über das CLI ist es möglich, Druckjobs und Drucker anzuhalten, Aufträge abubrechen und Druckjobs seitenweise aufzusetzen. Im Vergleich mit der Weboberfläche sind so mehr Funktionen erreichbar.

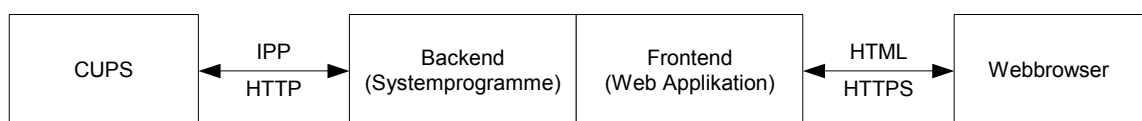
Aufbauend auf dieser Schnittstelle soll entsprechend dem Anforderungskatalog eine Web Applikation die Kommunikation mit dem Anwender übernehmen. Sie enthält die Mandantenverwaltung und stellt die einzige Oberfläche dar, mit der ein Anwender in Berührung kommt. Die Authentifizierung kann an zentraler Stelle durchgeführt werden.

Leider unterstützen auch die Kommandozeilenprogramme nicht alle gestellten Anforderungen, so daß auch hier zusätzliche Hilfsprogramme benötigt werden. Speziell das erneute Drucken nach einer Störung und das Umleiten auf einen anderen Drucker werden darüber nicht direkt unterstützt.

### 6.3.5 Variante 2b: Reimplementierung der Oberfläche und Kontrollprogramme

Eine Variante des vorangegangenen Ansatzes besteht darin, nicht die Kommandozeilenschnittstelle als Backend zu verwenden, sondern das Internet Printing Protocol einzusetzen. Es unterstützt neben der reinen Übertragung von Druckdaten auch viele Steuerkommandos. Darüber lassen sich alle im Anforderungskatalog festgelegten Kriterien erfüllen.

Die in CUPS enthaltene Bibliothek libcups kapselt das Internet Printing Protocol. Mit ihr ist es möglich, auf einfache Art und Weise aus einem C Programm heraus auf die Schnittstelle des Spoolers zuzugreifen. Darüber hinaus erfolgt eine Trennung zwischen Spooler und Verwaltungssoftware. Es ergibt sich eine „3 tier architecture“, wie sie in Abbildung 12 dargestellt ist.



**Abbildung 12: Aufbau Eigenentwicklung**

Wie bereits bei Variante 2a beschrieben, wird aufbauend auf den Kontrollprogrammen eine neue Bedienoberfläche implementiert. Die Kommunikation zwischen diesem Fron-

tend und den Kontrollprogrammen erfolgt dabei sehr direkt. Alle Verwaltungslogik wird von der Web Applikation übernommen, die Aufgabe der Kontrollprogramme besteht lediglich in der Kapselung der CUPS Funktionalität und Bereitstellung einfacher Kommandos. Diese können direkt vom Frontend bedient werden.

Durch dieses Vorgehen ist eine weitest gehend unabhängige Entwicklung möglich. Kontrollsoftware und Spooler sind voneinander entkoppelt. Änderungen auf einer der beiden Seiten beeinträchtigen nicht den Quelltext der anderen. Der höhere Entwicklungsaufwand dieser Lösung scheint dadurch gerechtfertigt, daß auf längere Sicht kein Abgleich verschiedener Entwicklungszweige erfolgen muß.

### 6.3.6 Fazit

Die vier aufgeführten Lösungsansätze verfolgen zwei unterschiedliche Vorgehensweisen. Variante 1 hat zum Ziel, große Teile des vorhandenen Codes wieder zu verwenden und über Modifikationen die gewünschten Funktionen zu integrieren. Sie hat den Vorteil, daß der Aufwand im Vergleich mit Variante 2 anfangs geringer ist. Bei Variante 2 wird versucht, über klare Schnittstellen auf den Spooler zuzugreifen. Durch die Entkoppelung der Eigenentwicklung kann von späteren Entwicklungen am Spooler profitiert werden, und es entstehen zwei unabhängige Programme. Dieser Ansatz erscheint insgesamt sinnvoller.

Es bleibt zu klären, welche Schnittstelle vorzuziehen ist. Das Kommandozeileninterface entspricht mit seinen Programmen dem des System V Spoolers. Es kann sehr einfach darauf aufgesetzt werden. Ein Zugriff über das Internet Printing Protocol stellt sich etwas komplexer dar. Das Protokoll muß zuerst gekapselt werden, um einen ähnlichen Komfort wie bei den vorhandenen Kommandozeilenprogrammen bereitzustellen. Da einige notwendige Funktionen jedoch nur so zu erreichen sind, muß zwangsläufig Variante 2b gewählt werden.

## **6.4 Implementierung**

### 6.4.1 Programmaufbau

Die neue Verwaltungssoftware untergliedert sich in zwei Bestandteile. Auf Systemseite übernimmt das Backend die Kommunikation mit dem Spooler. Es kapselt alle benötigten Funktionen und stellt diese über ein Kommandozeileninterface bereit. Das darauf aufbauende Frontend übernimmt die Kommunikation mit dem Webbrowser des Anwenders und enthält die höhere Programmlogik.

Da das Backend die Programmbibliothek libcups verwenden soll, um über das Internet Printing Protocol auf CUPS zuzugreifen und dafür ausschließlich C Header vorhanden sind, wird das Backend in C geschrieben. Mit der GNU Entwicklungsumgebung sind alle dazu benötigten Programme frei verfügbar: der GNU C Compiler (gcc), der GNU Debugger (gdb) und GNU Make.

Das Frontend soll als Web Applikation innerhalb des Apache Webservers laufen. Als Folge dessen wird es bei jedem Seitenzugriff neu gestartet und sämtliche Variablen sind zurückgesetzt. Über eine saubere Trennung zwischen Daten und Ansicht ist es jedoch möglich, die Datenhaltung persistent auszulegen. Dies geschieht zum einen über ein Sessionmanagement, das kurzzeitig relevante Daten vorhält, und zum anderen über das Backend, das den Zugriff auf Druckaufträge und die Benutzerverwaltung ermöglicht.

Wichtig für die Performance des Frontends ist daneben die Wahl der richtigen Programmiersprache. Der Apache Webserver ermöglicht auf der einen Seite das Ausführen von externen Programmen über das Common Gateway Interface (CGI). Diese müssen bei jedem Seitenaufruf neu gestartet werden und wirken sich somit negativ auf das Lastverhalten des Servers aus. Auf der anderen Seite unterstützt Apache über verschiedene Module die Skriptsprachen Perl, PHP und Python. Als Module bleiben sie zwischen den Seitenaufrufen im Speicher und müssen nicht erneut geladen werden. PHP erscheint dabei für die Entwicklung am besten geeignet. Neben der weiten Verbreitung, die gerade Außenstehenden einen leichten Zugang zum Quelltext ermöglicht, ist die Performance im Vergleich zu CGI Programmen wesentlich besser.

## 6.4.2 Backend

### 6.4.2.1 Kommunikation mit CUPS

Das Backend hat die Aufgabe, die Kommunikation mit CUPS zu kapseln. Die Eigenentwicklung baut hierzu eine IPP Verbindung zum Spooler auf und gibt eine Liste der vorgehaltenen Druckaufträge aus. Im nachfolgenden Auszug soll daher der Zugriff auf CUPS beispielhaft demonstriert werden. Eine weiterführende Beschreibung der Funktionen und Datenstrukturen ist im CUPS Programmierhandbuch enthalten und im CUPS Quelltext dokumentiert.

```
1: #include <stdlib.h>
2: #include <cups/cups.h>
3: #include <cups/language.h>
4:
5: int main(int argc, char *argv[])
6: {
7:     http_t *http = httpConnectEncrypt("localhost",
8:                                     ippPort(), cupsEncryption());
9:     cups_lang_t *lang = cupsLangDefault();
10:    ipp_t *request = ippNew(), *response;
11:    ipp_attribute_t *attr;
```

#### Abbildung 13: Kommunikation mit CUPS - Variablendefinition

In Zeile 7 wird eine HTTP-Verbindung zum Printspooler auf „localhost“ hergestellt. Als Portnummer und Verschlüsselung werden die Systemeinstellungen herangezogen. Für die Sprache wird ebenfalls der Default-Wert eingesetzt.

Innerhalb der HTTP-Verbindung wird das Internet Printing Protocol verwendet. Es wird dazu in Zeile 10 ein Container „request“ generiert, der alle Daten der Anfrage enthält. Die beiden Variablen „response“ und „attr“ werden erst später für die Auswertung der Serverantwort verwendet.

Abbildung 14 zeigt in den Zeilen 12 bis 20, wie die Anfrage vorbereitet wird. Es wird die gewünschte Funktion, im Beispiel „get jobs“, mit der „request\_id“ 1, angegeben. Zeichensatz, Sprache und URI werden festgelegt und mit der Funktion „ippAddString“ an die Anfrage angehängt. Je nach gewünschter Operation kann diese Liste beliebig erweitert werden.

```
12:     request->request.op.request_id = 1;
13:     request->request.op.operation_id = IPP_GET_JOBS;
14:
15:     ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_CHARSET,
16:         "attributes-charset", NULL, cupsLangEncoding(lang));
17:     ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_LANGUAGE,
18:         "attributes-natural-language", NULL, lang->language);
19:     ippAddString(request, IPP_TAG_OPERATION, IPP_TAG_URI,
20:         "printer-uri", NULL, "ipp://localhost/printers/");
21:
22:     if (!http)
23:         return EXIT_FAILURE;
24:     if (!(response = cupsDoRequest(http, request, "/")))
25:         return EXIT_FAILURE;
26:     if (response->request.status.status_code > IPP_OK_CONFLICT)
27:         return EXIT_FAILURE;
```

**Abbildung 14: Kommunikation mit CUPS - IPP Anfrage**

Nachdem die Anfrage komplett ist, wird ab Zeile 22 die Verbindung zum Server getestet, und HTTP Socket sowie IPP Container werden an die Funktion „cupsDoRequest“ übergeben. Dieser stellt die Serverantwort im Container „response“ bereit. Sollten währenddessen Fehler auftreten, wird das Programm mit dem Rückgabewert EXIT\_FAILURE verlassen.

```
28:     attr = response->attrs;
29:     while (attr) {
30:         if (attr->group_tag == IPP_TAG_JOB
31:             && !strcmp(attr->name, "job-name"))
32:             printf("%s\n", attr->values[0].string.text);
33:         attr = attr->next;
34:     }
```

**Abbildung 15: Kommunikation mit CUPS - Auswertung der Antwort**

Unter anderem enthält „response“ nach erfolgreicher Anfrage eine verkettete Liste mit Ergebnissen. Das darin enthaltene Feld „group\_tag“ gibt an, um welchen Typ es sich beim aktuellen Datensatz „attr“ handelt. Der Programminterne Name ist dem Feld „name“ zu entnehmen. „values“ enthält wiederum eine Liste, in welcher die zugehörigen

Werte enthalten sind. Im beschriebenen Beispiel hat diese Liste immer die Länge eins. Es kann so, wie in Zeile 32 gezeigt, direkt auf diesen Wert zugegriffen werden.

Die Datenstruktur `ipp_t`, von deren Typ auch „response“ ist, enthält neben den gezeigten Feldern noch weitere. Als zentrale Datenstruktur verbindet sie viele der definierten „structs“ und „unions“ und erreicht eine fünfstufige Hierarchieebene. Eine vollständige Beschreibung der Attribute ist nur in den Headerdateien und im CUPS Quelltext zu finden.

```
35:     httpClose (http) ;
36:     cupsLangFree (lang) ;
37:     ippDelete (request) ;
38:     ippDelete (response) ;
39:
40:     return EXIT_SUCCESS ;
41: }
```

### Abbildung 16: Kommunikation mit CUPS - Speicher aufräumen

Abschließend werden in den Zeilen 35-38 die dynamisch belegten Speicherbereiche freigegeben und das Programm beendet sich mit dem Return Code „EXIT\_SUCCESS“.

#### 6.4.2.2 Funktionsumfang des Backends

Die im Backend enthaltenen Funktionen sind über sechs separate Programme erreichbar. Dies dient in erster Linie der Gliederung und dem leichteren Verständnis bei der manuellen Bedienung der Eigenentwicklung. Die nach außen abgebildeten Funktionen werden vom Backend dazu in mehrere einzelne IPP Kommandos übersetzt und an den Spooler weitergereicht. Ebenso werden die Serverantworten gefiltert, um in höheren Ebenen nicht mehr benötigte Daten möglichst früh auszublenden.

Die nachfolgende Auflistung enthält eine Beschreibung dieser sechs Programme und den mit ihnen realisierten Funktionen:

- Mit `getjobs` kann eine Liste aller Druckaufträge einschließlich deren Attribute abgefragt werden. Diese Liste kann über Parameter auf eine einzelne Drucker eingeschränkt werden. In ihr sind Auftragsnummer und `-name`, Druckername, Statuscode, Größe, Erst- und Druckdatum enthalten.
- `getprinters` liefert Details aller Drucker einer Klasse und deren Attribute. In dieser Liste sind neben dem Druckernamen, Statuscode, Statusbeschreibung, Stand-

ort- und Modellbeschreibung und die Zahl der Aufträge in der Warteschlange enthalten.

- Über `setjob` können Druckaufträge verändert werden. Es ist möglich, noch nicht gedruckte Jobs anzuhalten und wieder fortzusetzen. Laufende Aufträge können abgebrochen werden.
- Abgebrochene oder erfolgreich abgeschlossene Aufträge können mit `restartjob` erneut gestartet werden. Dabei ist es möglich, diese an einen anderen Drucker umzuleiten und den Druckbereich seitengenau einzuschränken.
- Das Programm `setprinter` dient dazu, den Status eines Druckers oder dessen Warteschlange zu verändern. Drucker und die zugehörige Warteschlange können angehalten und wieder gestartet werden.
- `purgejob` verschiebt Druckaufträge in einen „Papierkorb“. Dies ist intern ein spezieller Drucker für jeden Mandanten, der nur dem Administrator zugänglich ist und das Wiederherstellen gelöschter Aufträge ermöglicht.

### 6.4.2.3 Kommunikationsschnittstelle

Die Kontrollprogramme des Backends sind über ein Kommandozeileninterface erreichbar. Dies hat gegenüber einer Programmbibliothek den Vorteil, daß die Programme darüber auch manuell bedient werden können. Dies bietet einen guten Aufsetzpunkt bei der Fehlersuche. Dementsprechend sind die zurückgegebenen Daten auch für den Menschen lesbar formatiert und nicht direkt als PHP Datenstrukturen aufbereitet.

```
PS-Farbe-Test

printer-state == 5
printer-state-message == Paused
printer-is-accepting-jobs == 1
queued-job-count == 0
printer-name == PS-Farbe-Test
printer-location == Finger1 Erdgeschoss dsb AG
printer-info == Tektronix Farbdrucker
color-supported == 1
printer-make-and-model == Generic postscript color printer
```

**Abbildung 17: Programmausgabe "getprinters demouser"**

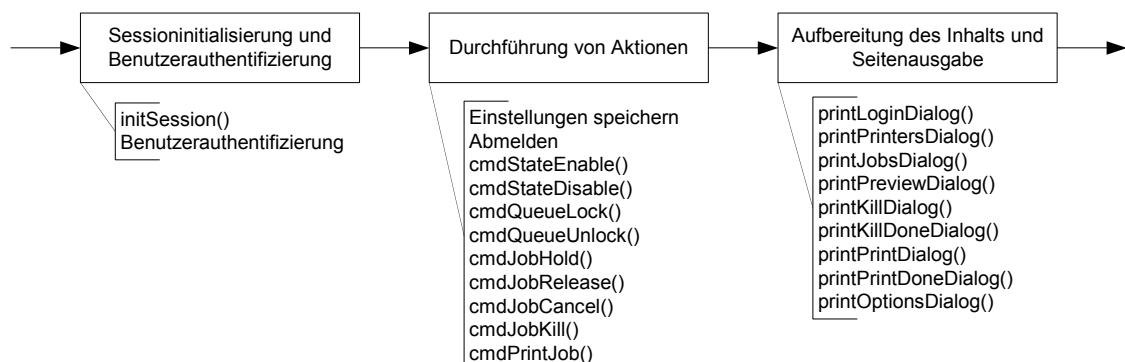
Abbildung 17 zeigt die Ausgabe des Kommandos `getprinters demouser`. Diesem Anwender ist ein Drucker mit dem Namen „PS-Farbe-Test“ zugeordnet. Der Drucker ist momentan angehalten („printer-state == 5“). Es befinden sich keine Aufträge in der Warteschlange („queued-job-count == 0“), die Annahme neuer Jobs wird akzeptiert („printer-is-accepting-jobs == 1“). Weiter unterstützt der Drucker Farbausgabe („color-supported == 1“), und aus der Modellbeschreibung geht hervor, daß es sich um einen PostScript Drucker handelt.

### 6.4.3 Frontend

#### 6.4.3.1 Aufbau

Das Frontend des Print Output Systems ist wie das Backend eine Eigenentwicklung. Es ist vollständig in PHP programmiert und erzeugt dynamisch XHTML/CSS Seiten, die die Programmausgabe im Webbrowser des Anwenders darstellen. Die gesamte Kommunikation wird währenddessen durch eine HTTPS-Verbindung geschützt. Sie stellt gleichermaßen die Authentizität des Print Output Systems gegenüber dem Anwender sicher und verschlüsselt persönliche Daten während der Übertragung.

Der Aufbau des Frontends, wie in Abbildung 18 dargestellt, gliedert sich in drei unterschiedliche Funktionsbereiche. Bei einem Seitenaufruf erfolgt zunächst die Initialisierung der Session und die Rekonstruktion alter Verbindungsdaten. Nach erfolgreicher Benutzerauthentifizierung werden Aktionen durchgeführt und es erfolgt die Aufbereitung des Inhaltes. Im Anschluß wird die erzeugte Seite ausgegeben und auf die nächste Benutzereingabe gewartet.



**Abbildung 18: Aufbau des Frontends**

Dieser Aufbau läßt sich in die Bereiche Eingabe, Verarbeitung und Ausgabe zerlegen, wie sie das klassischen EVA Prinzip beschreibt. Vorteilhaft an der Trennung von Verarbeitung und Ausgabe ist die hohe Flexibilität. Aktionen können dort eingebunden werden, wo sie der Anwender wünscht und sind losgelöst von einzelnen Dialogen. Der Bereich Eingabe wurde bei dieser Implementierung um Sessionmanagement und Benutzerauthentifizierung erweitert. Dies ist im klassischen EVA Modell so nicht vorgesehen, macht bei einer Web Applikation jedoch Sinn. Es besteht kein Unterschied zwischen neuen Benutzereingaben und älteren, die in einer Session abgelegt wurden.

Funktionsname	Beschreibung
cmdStateEnable()	Drucker starten
cmdStateDisable()	Drucker anhalten
cmdQueueLock()	Warteschlange sperren
cmdQueueUnlock()	Warteschlange freigeben
cmdJobHold()	Verarbeitung des Auftrages anhalten
cmdJobRelease()	Verarbeitung des Auftrages fortsetzen
cmdJobCancel()	Verarbeitung des Auftrages abbrechen
cmdJobKill()	Auftrag löschen
cmdPrintJob()	Auftrag drucken

**Tabelle 6: Frontend - Verarbeitungsfunktionen**

Eine Liste mit Funktionen aus dem Verarbeitungsbereich ist in Tabelle 6 aufgeführt. Es handelt sich dabei lediglich um die Hauptfunktionen, eventuelle Unterfunktionen sind an dieser Stelle nicht von Bedeutung. Tabelle 7 enthält eine entsprechende Funktionsbeschreibung der Ausgabefunktionen. Der Zusammenhang zwischen den einzelnen Dialogen und ihr genauer Funktionsumfang wird im nächsten Abschnitt „Arbeiten mit dem Frontend“ geklärt.

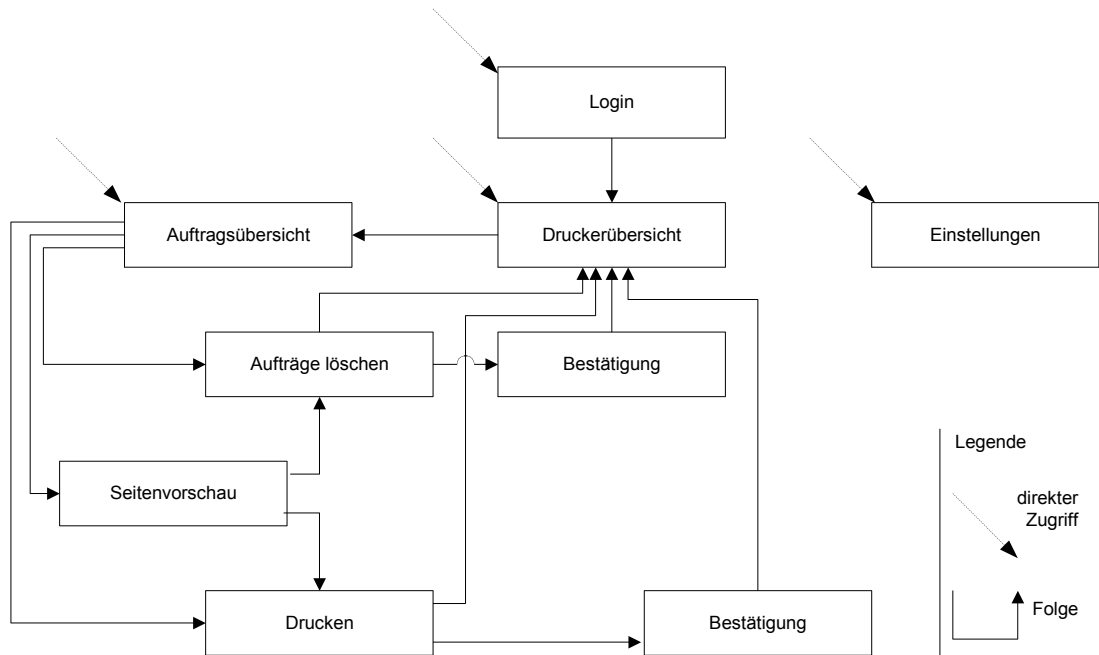
Funktionsname	erzeugte Seite
printLoginDialog()	Anmeldefenster
printPrintersDialog()	Druckerübersicht
printJobsDialog()	Auftragsübersicht
printPreviewDialog()	Seitenvorschau
printKillDialog()	Löschen: Rückfrage
printKillDoneDialog()	Löschen: Aktion wurde durchgeführt
printPrintDialog()	Drucken: Auswahl von Druckoptionen
printPrintDoneDialog()	Drucken: Aktion wurde durchgeführt
printOptionsDialog()	Einstellungen

**Tabelle 7: Frontend - Eingabefunktionen**

#### 6.4.3.2 Arbeiten mit dem Frontend

Bei der Entwicklung der Web Applikation wurden stets die Ziele des Anforderungskataloges verfolgt. Die Funktionalität wurde um die beiden Dialoge „Auftragsübersicht“ und „Druckerübersicht“ angeordnet. Der Aufbau dieser beiden Dialoge ist an die CUPS Weboberfläche angelehnt. Von dort können sämtliche Funktionen des Frontends erreicht werden. Abbildung 19 zeigt die Dialogabfolge und die Pfade, denen ein Anwender während der Bedienung folgen kann. Fortfolgend werden die einzelnen Dialoge

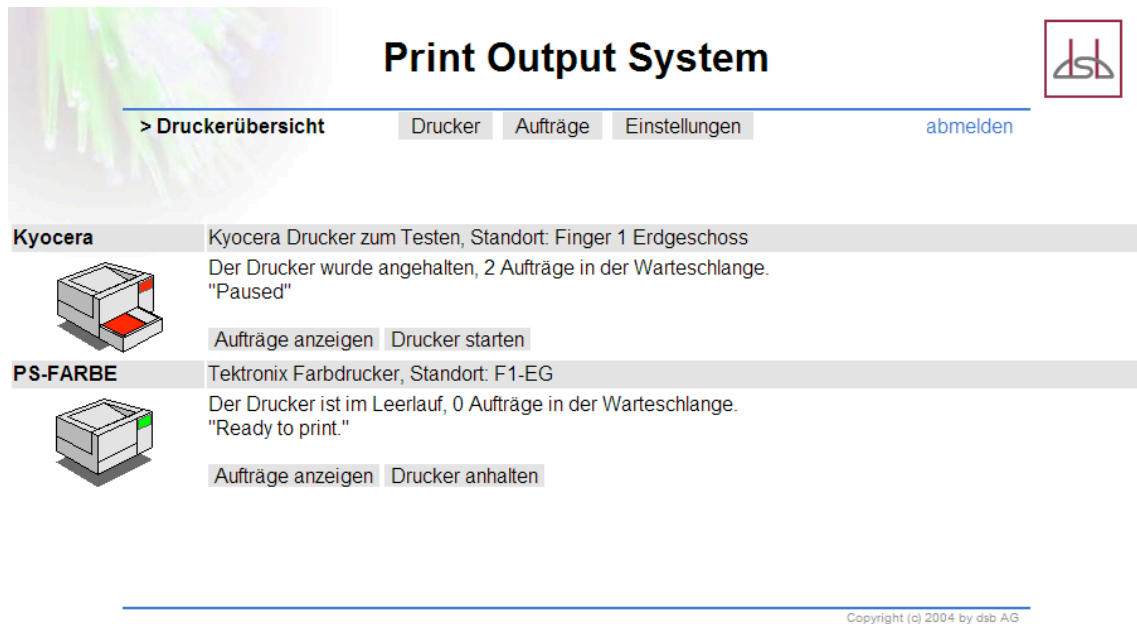
und die damit assoziierten Funktionen gezeigt. Eine Illustration sämtlicher Dialoge ist dem Anhang zu entnehmen.



**Abbildung 19: Frontend - Dialogabfolge**

Zu Beginn einer Sitzung wird der *Anmeldedialog* angezeigt. Der Anwender muß sich an dieser Stelle mit Benutzernamen und Kennwort authentifizieren. Spracheinstellungen und zuletzt verwendeter Benutzername werden aus einem clientseitig gespeicherten Cookie übernommen, um die Oberfläche so angenehm wie möglich zu gestalten. Nach erfolgreicher Authentifizierung erfolgt der Übergang zur Druckerübersicht.

Die *Druckerübersicht* (Abbildung 20) stellt den zentralen Einstiegspunkt der Oberfläche dar. Über das Menü im oberen Bereich können die Dialoge Druckerübersicht, Auftragsübersicht und Einstellungen zu jeder Zeit direkt erreicht werden. Der Link „abmelden“ beendet die aktuelle Session und führt zurück zum vorangegangenen Anmeldedialog.



**Abbildung 20: Frontend - Druckerübersicht**

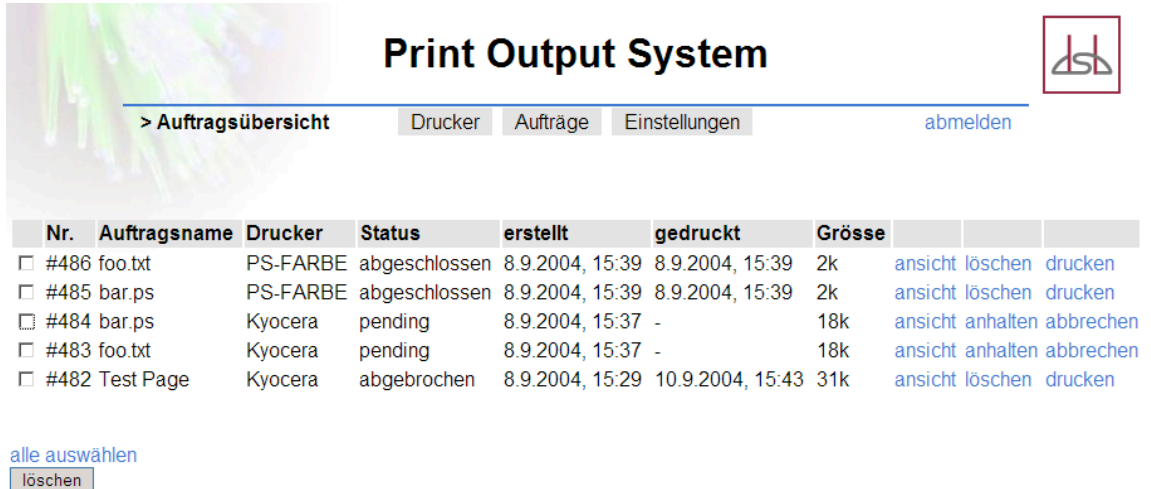
Im Mittelteil ist eine Liste aller für diesen Benutzer verfügbaren Drucker abgebildet. Der Drucker „Kyocera“ wurde angehalten; es befinden sich noch zwei Aufträge in der Warteschlange. Die grau hinterlegte Schaltfläche „Drucker starten“ bewirkt ein Fortsetzen der Verarbeitung ausstehender Druckaufträge. Entsprechend verhält es sich mit der Schaltfläche „Drucker anhalten“ des Druckers „PS-Farbe“; durch einen Click wird die Druckausgabe ausgesetzt.

Folgt man der Schaltfläche „Aufträge anzeigen“, gelangt man in die *Auftragsübersicht* (Abbildung 21). Dort zeigt eine Liste alle Aufträge des Benutzers beziehungsweise des aktuell ausgewählten Druckers. Für jeden Druckauftrag sind seine eindeutige Identifikationsnummer, Auftragsame, Drucker, Status, Erstell- und Druckdatum sowie die Größe angegeben. Hier können die Aufträge angezeigt und je nach Status gelöscht, angehalten, fortgesetzt, abgebrochen oder erneut gedruckt werden.

Die Verknüpfung „ansicht“ führt weiter zur *Seitenvorschau*. Sie ermöglicht die Ansicht von Druckaufträgen im Spooler. Die Daten werden seitenweise als JPEG Grafiken dargestellt und lassen sich über zwei Schaltflächen vor- bzw. zurückblättern. Von dieser Stelle aus können die Dialoge Löschen und Drucken erreicht werden.

Über die Schaltfläche „löschen“ wird der entsprechende Dialog *Löschen* aufgerufen. Die markierten Aufträge werden nochmals aufgelistet und nach Bestätigung in den „Papierkorb“ verschoben. Dort sind sie für den Anwender nicht mehr sichtbar, können aber vom Administrator wiederhergestellt werden.

Über den Dialog *Drucken* können Aufträge mit den vorherigen Einstellungen erneut ausgegeben werden. Weiter können sie auch auf einen anderen Drucker umgelenkt oder seitenweise aufgesetzt werden. Dies ist speziell im Fehlerfall interessant, wenn beispielsweise kein Toner mehr vorhanden ist oder ein Papierstau vorliegt und durch den Nachdruck unnötige Nachdruckzeiten vermieden werden sollen.



The screenshot shows the 'Print Output System' web interface. At the top, there is a navigation bar with the title 'Print Output System' and a logo on the right. Below the title, there are tabs for '> Auftragsübersicht', 'Drucker', 'Aufträge', and 'Einstellungen', along with a link 'abmelden'. The main content area displays a table of print jobs with columns for 'Nr.', 'Auftragsname', 'Drucker', 'Status', 'erstellt', 'gedruckt', and 'Grösse'. Each row includes a checkbox and a set of action links: 'ansicht', 'löschen', and 'drucken' or 'abbrechen'.

Nr.	Auftragsname	Drucker	Status	erstellt	gedruckt	Grösse			
<input type="checkbox"/> #486	foo.txt	PS-FARBE	abgeschlossen	8.9.2004, 15:39	8.9.2004, 15:39	2k	<a href="#">ansicht</a>	<a href="#">löschen</a>	<a href="#">drucken</a>
<input type="checkbox"/> #485	bar.ps	PS-FARBE	abgeschlossen	8.9.2004, 15:39	8.9.2004, 15:39	2k	<a href="#">ansicht</a>	<a href="#">löschen</a>	<a href="#">drucken</a>
<input type="checkbox"/> #484	bar.ps	Kyocera	pending	8.9.2004, 15:37	-	18k	<a href="#">ansicht</a>	<a href="#">anhalten</a>	<a href="#">abbrechen</a>
<input type="checkbox"/> #483	foo.txt	Kyocera	pending	8.9.2004, 15:37	-	18k	<a href="#">ansicht</a>	<a href="#">anhalten</a>	<a href="#">abbrechen</a>
<input type="checkbox"/> #482	Test Page	Kyocera	abgebrochen	8.9.2004, 15:29	10.9.2004, 15:43	31k	<a href="#">ansicht</a>	<a href="#">löschen</a>	<a href="#">drucken</a>

[alle auswählen](#)

Copyright (c) 2004 by dsb AG

### Abbildung 21: Frontend - Auftragsübersicht

Die Links „löschen“, „anhalten“, „fortsetzen“ und „abbrechen“ aus der *Auftragsübersicht* führen diese Aktionen direkt aus. Sie bleiben im aktuellen Dialog stehen und haben keinen eigenen Folgedialog definiert.

Der dritte Eintrag im Hauptmenü „Einstellungen“ führt zum *Einstellungsdialog*. Dieser dient bisher einzig zur Wahl der Oberflächensprache. Es kann zwischen Deutsch, Englisch und Französisch entschieden werden. Die diesbezügliche Konfiguration wird, wie eingangs bereits beschrieben, in einem Cookie auf dem Client gespeichert und steht für die Verweildauer des Cookies zur Verfügung.

## 7 Resümee

Für den Aufbau eines neuen Print Output Systems erwies sich die Kombination von Open Source Programmen und eigenem Programmcode als ein sehr guter Ansatz. Das Common Unix Printing System enthält die Funktionen, wie sie von unseren Kunden gewünscht werden. Fehlende Funktionalität konnte leicht in das Gesamtsystem integriert werden und es entstand eine Lösung, die optimal an die gestellten Anforderungen angepaßt ist.

Die durchgängige Verwendung standardisierter Schnittstellen erlaubt es, einzelne Komponenten auszutauschen. Neue Funktionalität und Fehlerkorrekturen an der zugrunde liegenden Fremdsoftware kann so problemlos in das Print Output System integriert werden. Kostenintensive Anpassungen mit jedem Versionswechsel bei CUPS, GhostScript oder anderen eingesetzten Programmen sind damit hinfällig.

Mit der Entscheidung, das User Interface des Print Output Systems als Web Application auszulegen, ergeben sich einige Vorteile bei der Programmpflege. So ist es nicht nötig ein stabiles Release zu erstellen und an unsere Kunden auszuliefern. Die aktuelle Version im Haus ist stets das stabile Release. Liegt ein Problem vor, kann dies leicht nachvollzogen werden. Fehlerkorrekturen sind sofort systemweit verfügbar – häufig bevor der Fehler ein zweites Mal bemerkt wird.

Neben den technischen Vorteilen, die sich aus dem Einsatz freier Software ergeben, ist der Einsatz von Open Source Programmen wirtschaftlich interessant. Ein Application Service Provider, wie die dsb AG, wird von den restriktiven Lizenzbestimmungen der GNU General Public License nur wenig berührt. Da ein ASP lediglich die Programmausgabe, nicht aber das eigentliche Programm, weitergibt, besteht für ihn kein Zwang eigene Versionen zu veröffentlichen.

# Anhang

## A Ergänzende Screenshots

### VTAMPRINT

```

VTAMPRINT 4520G                ** Signon **                11/08/2004 16:01:47
                                                                    Terminal:  BDSG30D0

**      ** *****          ****      **      ** *****          *****          *****          **      ** *****
**      ** *****          **      **      **      **      **      **      **      **      **      **      ** *****
**      **      **      **      **      **      **      **      **      **      **      **      **      **      **
**      **      **      **      **      **      **      **      **      **      **      **      **      **      **
**      **      **      *****          **      **      **      **      **      **      **      **      **      **
**      **      **      **      **      **      **      **      **      **      **      **      **      **      **
**      **      **      **      **      **      **      **      **      **      **      **      **      **      **
**      **      **      **      **      **      **      **      **      **      **      **      **      **      **

                Software Excellence from Macro 4
                (c) Copyright Macro 4 1999. All rights reserved

                User id ==> █

Type your VTAMPRINT USER-ID and press ENTER
Pfk1      2      3 QUIT      4      5      6
Pfk7      8      9      10      11      12
    
```

Abbildung 22: VTAMPRINT - Login

```

VTAMPRINT 4520G                Primary Option Menu          11/08/2004 16:06:27
User:  IPVX                                                                    Terminal:  BDSG30D0

                1  DPRT  Printer Display & Control
                2  DLST  POWER List Queue Display
                3  DPUN  POWER Punch Queue Display
                4  DRDR  POWER Reader Queue Display
                5  DSUBS Subset Display
                6  DPCP  PCPrint Default Printer Definition
                7  DLOG  Audit Display
                8  DREQ  Print requests

                Q  QUIT  Quit From VTAMPRINT
                H  HELP  Help

                N  NEW   What's New in VTAMPRINT

                Software Excellence from Macro 4
                (c) Copyright Macro 4 1999. All rights reserved

COMMAND ==> █
Select option by cursor, option number/letter or command and press Enter.
Pfk1      2      3 QUIT      4      5      6
Pfk7 UP    8 DOWN  9      10 VIEW  11      12
    
```

Abbildung 23: VTAMPRINT - Hauptmenü

```
VTAMPRINT 4520G          Printer Display & Control          11/08/2004 16:09:20
User:  IPVX              Terminal:  BDSG30D0
CMD  Printer  -Status-  Subset  Cl  Def-FCB  Forms  Jobname  Jobno  Pages  Copy_
-----*-----
      IPVEDV   LEERLAUF  S109   R           3751
      IPVVERT  LEERLAUF  S143   R           1159

COMMAND ==>
CMD: Start stop setUp Go Halt restart(+nn/nn) Flush retrieve(?) eXtra SHIFT
Pfk1      2      3 QUIT      4      5      6
Pfk7 UP    8 DOWN    9      10 VIEW   11     12
```

Abbildung 24: VTAMPRINT - Druckerübersicht

### Frontend

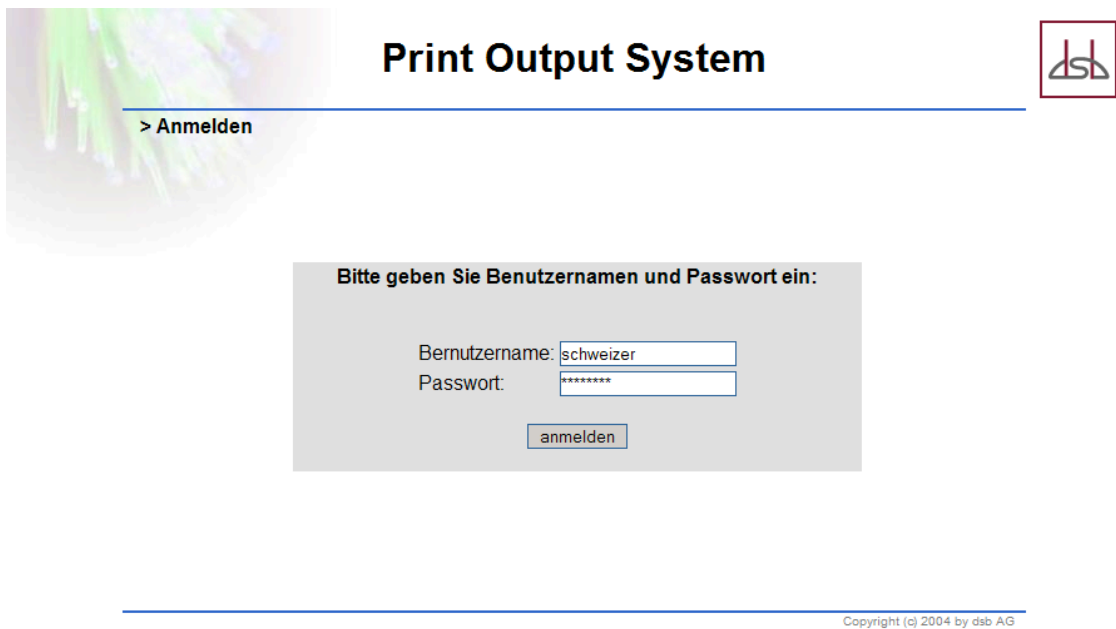


Abbildung 25: Frontend - Login



Abbildung 26: Frontend - Druckeigenschaften

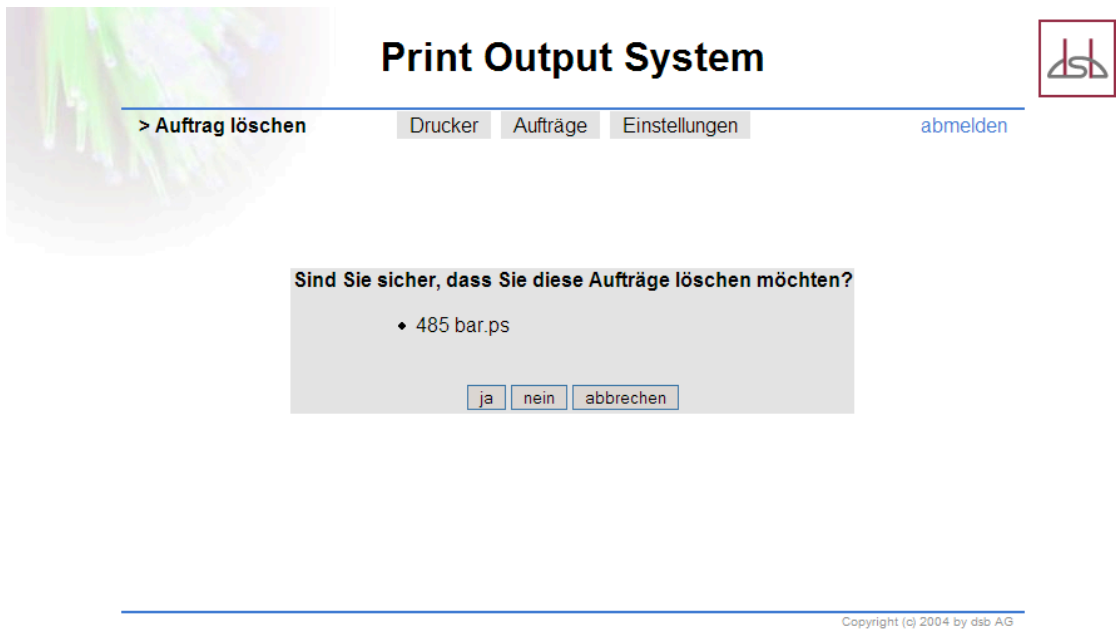
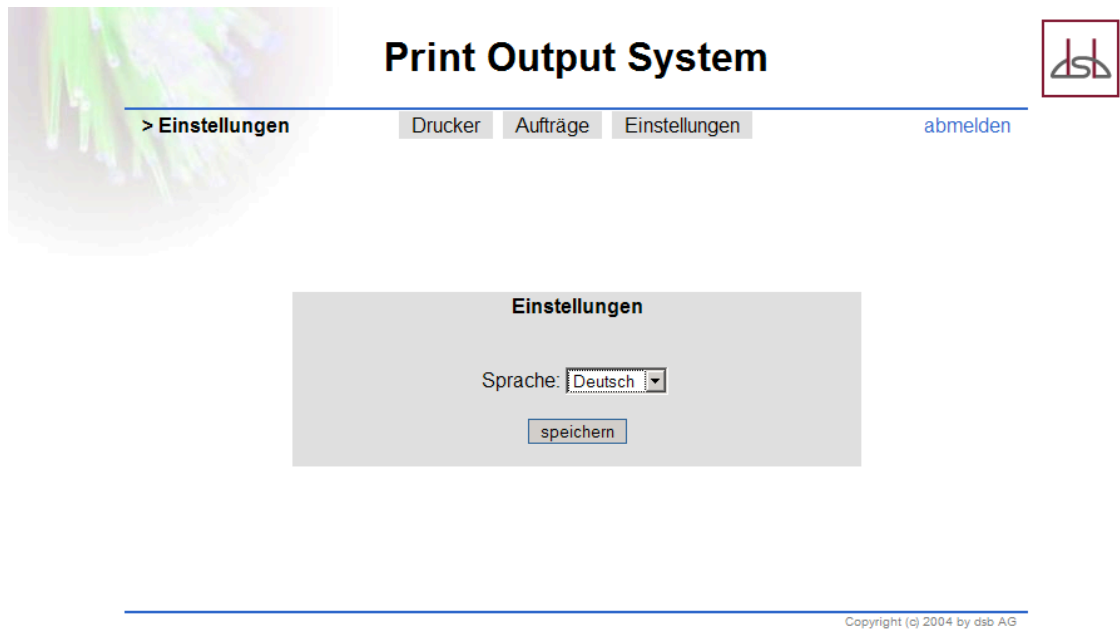


Abbildung 27: Frontend - Auftrag löschen



**Abbildung 28: Frontend - Einstellungen**

## ***B Überblick des Quelltextes***

### Backend

<b>Dateiname</b>	<b>Beschreibung</b>
Makefile	GNU Make Kommandos
getjobs.c	Erstellt Liste aller Aufträge einer Klasse
getprinters.c	Erstellt Liste aller Drucker einer Klasse
ipp.c	Gemeinsame Funktionen zur IPP-Einbindung
ipp.h	C Header Datei
jobs.c	Gemeinsame Funktionen zur Job-Manipulation
jobs.h	C Header Datei
printers.c	Gemeinsame Funktionen zur Drucker-Manipulation
printers.h	C Header Datei
purgejob.c	Löschen von Aufträgen
restartjob.c	Wiederaufsetzen von Aufträgen
setjob.c	Manipulation von Aufträgen
setprinter.c	Manipulation von Druckern

**Tabelle 8: Quelltextverzeichnis Backend**

### Frontend

<b>Verzeichnis/Dateiname</b>	<b>Beschreibung</b>
index.php	Symbolischer Link auf das Hauptprogramm
pops.php	Hauptprogramm
lib/	PHP Programmteile
lib/conf/	Konfiguration
lib/core/	Zwingend notwendige Komponenten
lib/functions/	Kapselung des Backends, höhere Funktionen
lib/dialogs/	Dialoge und Ausgabefunktionen
lib/lang/	Internationalisierung
lib/css/	Cascading Style Sheets
lib/img/	Bilder (statisch)
lib/img/preview.php	Dynamische Bildgenerierung für Vorschau

**Tabelle 9: Quelltextverzeichnis Frontend**

```
<?php
/*
 * preview.php (c) 2004 by dsb AG, http://www.dsb.net/
 *
 * Benjamin Schweizer <schweizer@dsb.net>
 *
 * *****/

/*
 * imagemagick wrapper
 */

include ('../lib/conf/config.inc');
include ('../lib/functions/auth.inc');
include ('../lib/functions/file.inc');
include ('../lib/functions/jobs.inc');
include ('../lib/functions/printers.inc');

header("Content-type: image/jpeg");
session_start();
$user = $_COOKIE['user'];
$pass = $_SESSION['pass'];

$id = $_GET['id']; $page = $_GET['page'];
$file = $spoolDir . '/' . getJobFile($id);
$tempfile = '/tmp/cupspreview-' . $user . '.jpg';
if (hasJobPermission($id) and is_numeric($page)) {
    switch (fileGuess("$file")) {
        case "application/postscript":
        case "application/pdf":
            `convert $file\[ $page\] -scale 800x800 $tempfile`;
            if (`ls $tempfile`) {
                echo `cat $tempfile`;
            } else {
                echo `cat ./en_pna.jpg`; // preview not available
            }
            break;
    }
}
```

```
case "text/plain":
    // 842x595 == A4.Landscape
    `convert -page 842x595 -font courier \
        text:$spoolDir/$file\[ $page\] - scale 800x800 \
        $tempfile`;
    if (`ls $tempfile`) {
        echo `cat $tempfile`;
    } else {
        echo `cat ./en_pna.jpg`; // preview not available
    }
    break;
default:
    echo `cat ./en_pna.jpg`;
}
`rm $tempfile`;
} else {
    echo `cat ./en_ad.jpg`; // access denied
}

?>
```

Abbildung 29: Frontend - preview.php

**C CD-ROM**

Verzeichnis	Beschreibung
/source/backend/	Quelltext des Backends
/source/frontend/	Quelltext des Frontends
/source/sample/	Quelltext des Beispielprogrammes aus Kapitel 6.4.2.1
/screenshots/vtamprint/	Screenshots von VTAMPRINT
/screenshots/frontend/	Screenshots der Eigententwicklung
/licenses/	Lizenztexte der GNU General Public License, BSD License und Artistic License
/other/bsd-lpr/	Quelltext BSD Line Printer 2004-09-23
/other/ceps/	Quelltext Cisco Enterprise Print System 2.24
/other/cps/	Quelltext Coherent Printing System 2003-11-28
/other/cups/	Quelltext Common Unix Printing System 1.2 CVS 2004-09-26
/other/foomatic/	Datenbank und Skripte Foomatic 3.0 2004-09-26
/other/gimp-print/	Quelltext Gimp-Print 2.4.7
/other/gnulpr/	Quelltext GNU Line Printer 1.0.1
/other/hpijs/	Quelltext HP InkJet Server 1.6.2
/other/ifhp/	Quelltext ifhp 3.5.18
/other/lprng/	Quelltext LPRng 3.8.28
/other/lprng-tool/	Quelltext LPRng Tool 1.3.2
/other/omni/	Quelltext OMNI Printer Driver Model 0.9.1
/other/pdq/	Quelltext Print Don't Queue 2.2.1
/other/ppr/	Quelltext Post Script Printer 1.52

**Tabelle 10: Inhalt der CD-ROM**

## Quellenverzeichnis

### Bücher

- [Ado99] Adobe Systems Inc.: *PostScript Language Reference*, 3. Auflage. Addison-Wesley, 1999 – (ISBN 0-201-37922-8)
- [Bal00] Balzert, Helmut: *Lehrbuch der Software-Technik*, 2. Auflage. Spektrum Akademischer Verlag, 2000 – (ISBN 3-8274-0480-0)
- [Con94] McConnell, Steve: *Code complete*. Microsoft Press, 1994 – (ISBN 3-86063-333-3)
- [Gra04] Graham, Paul: *Hackers & Painters*. O'Reilly, 2004 – (ISBN 0-596-00662-4)
- [Kof98] Kofler, Michael: *Linux – Installation, Konfiguration, Anwendung*, 3. Auflage. Addison-Wesley, 1998 – (ISBN 3-8273-1304-X)
- [Kof02] Kofler, Michael: *Linux – Installation, Konfiguration, Anwendung*, 6. Auflage. Addison-Wesley, 2002 – (ISBN 3-8273-1854-8)
- [Kyo98] FS-3700+ Anwenderhandbuch, Version 2. Kyocera Electronics Europe GmbH, Mollsfeld 12, Meersbusch/Deutschland, 1998 – (keine ISBN)
- [Mui96] Mui, Linda: *Wenn der Unix-System-Administrator nicht zu finden ist*. O'Reilly/International Thomson Verlag GmbH & Co KG, 1996 – (ISBN 3-930673-23-1)
- [Pit00] Pitts, David u. Ball, Bill: *Red Hat Linux 6 Kompendium*. Markt & Technik Verlag, 2000 – (ISBN 3-8272-5684-4)
- [Red99] *Red Hat Linux 7 Referenzhandbuch*. Red Hat, Inc., 1999 – (keine ISBN)
- [Reh03] Rehman, Rafeeq Ur u. Paul, Christopher: *The Linux Development Platform*. Pearson Education, Inc, 2003 – (ISBN 0-13-009115-4)
- [Rei90] Reid, Glenn C.: *Thinking in PostScript*. Addison-Wesley, 1990 – (ISBN 0-201-52372-8)
- [Roc85] Rochkind, M.: *Advanced Unix Programming*. Prentice Hall, 1985 – (ASIN 0130118001)
- [Sch97] Schüler, Nikolaus: *Der Gcc-Compiler*. bhv Verlags GmbH, 1997 – (ISBN 3-98360-873-7)
- [Sus01] *SuSE Linux – das Handbuch*, 19. Auflage. SuSE GmbH, 2001 – (ISBN 3-934678-25-4)
- [Swe01] Sweet, Michael: *CUPS: Common Unix Printing System*. SAMS, 2001 – (ISBN 0-672-32196-3)

- [Sto00] Stoll, Rolf D. und Leirer, Gudrun Anna: *PHP4 + MySQL*, 2. Auflage. Data Becker, 2000 – (ISBN 3-8158-2043-X)
- [Tan01] Tanenbaum, Andrew S.: *Modern Operating Systems*, 2. Auflage. Prentice-Hall, Inc., 2001 – (ISBN 0-13-031358-0)
- [Tor01] Torvalds, Linus: *Just for Fun*. HarperCollins Publisher, Inc., 2001 – (ISBN 3-446-21684-7)
- [Wie95] Wielsch, Michael: *Das große Buch zu Unix*. Data Becker, 1995 – (ISBN 3-8158-1166-X)
- [Xep98] *The Handbook of IBM Terminology*. Xephon plc, 27-35 London Road, Newbury/England, 1998 – (keine ISBN)

### Online Quellen

- [Cha04] Chappell, David: *The PPR Spooler*. Letzte Aktualisierung am 20. Februar 2004 – (<http://ppr.trincoll.edu/>)
- [ESP03] Easy Software Products: *CUPS Software Programmers Manual, Version 1.1.20*. Abfrage v. 13. September 2004 – (<http://www.cups.org/spm.html>)
- [Esp04] Easy Software Products: Common Unix Printing System. Abfrage v. 26. September 2004 – (<http://www.cups.org/>)
- [FSF04] Free Software Foundation: *Frequently Asked Questions about the GNU GPL*. Letzte Aktualisierung am 11. Mai 2003 – (<http://www.gnu.org/licenses/gpl-faq.html>)
- [FSF91] Free Software Foundation: *GNU General Public License Version 2*. Veröffentlicht im Juni 1991 – (<http://www.gnu.org/copyleft/gpl.html>)
- [Gim04] Gimp-Print Project Team: Gimp-Print. Letzte Abfrage vom 26. September 2004 – (<http://gimp-print.sourceforge.net/>)
- [HPC04] Hewlett-Packard Company: HP Linux InkJet Driver Project. Letzte Aktualisierung am 17. September 2004 – (<http://hpinkjet.sourceforge.net/>)
- [Ive04] Ivereigh, Damian: Cisco Enterprise Print System. Letzte Aktualisierung am 11. März 2004 – (<http://ceps.sourceforge.net/>)
- [Kam04] Kampeter, Till und Taylor, Grand: Foomatic. Letzte Abfrage vom 26. September 2004 – (<http://www.linuxprinting.org/foomatic.html>)
- [Kom00] Komarnitsky, Alek: LPRng lpinfo. Letzte Aktualisierung am 6. März 2000 – (<http://www.komar.org/pres/lpinfo/>)

- [Kpf02] Pfeifle, K.: *Tutorial on CUPS and Foomatic*. Letzte Aktualisierung im September 2002 – (<http://www.linuxprinting.org/kpfeifle/LinuxKongress2002/Tutorial/>)
- [Lai90] McLaughlin III, Leo J: *RFC 1179: Line Printer Daemon Protocol*. Letzte Aktualisierung im August 1990 – (<ftp://ftp.rfc-editor.org/in-notes/rfc1179.txt>)
- [Lan00] Langford, Jacob A.: *Print Don't Queue*. Letzte Aktualisierung am 31. März 2000 – (<http://pdq.sourceforge.net/>)
- [Lev04] Lévénéz, Éric: *Unix History*. Letzte Aktualisierung am 10. September 2004 – (<http://www.levenez.com/unix/>)
- [LTC03] IBM Linux Technology Center: *OMNI Printer Driver Model*. Letzte Aktualisierung am 19. Dezember 2003 – (<http://www-124.ibm.com/developerworks/oss/linux/projects/omni/>)
- [Maj03] Majer, Adam: *Debian Package of LPR*. Letzte Aktualisierung am 23. September 2003 – (<http://packages.debian.org/unstable/net/lpr>)
- [OSI04] Open Source Initiative: *The Open Source Definition, Version 1.9*. Abfrage vom 13. September 04 (<http://www.opensource.org/docs/definition.php>)
- [Pow04] Powell, Patrick: *Next Generation Line Printer*. Letzte Aktualisierung am 23. September 2004 – (<http://www.lprng.com/>)
- [Woo01] Woodard, Ben: *GNU Line Printer*. Letzte Aktualisierung am 4. Dezember 2001 – (<http://lpr.sourceforge.net/>)
- [Wor03] Worthington, Thomas: *Coherent Printing System*. Letzte Aktualisierung am 28. November 2003 – (<http://www.tww.cx/cps.php>)