

DIPLOMARBEIT

Robert Lukaschek

**Konzeption und Entwicklung eines
Personalinformationssystems**

2010



DIPLOMARBEIT

Konzeption und Entwicklung eines Personalinformationssystems

Autorin: **Robert Lukaschek**

Studiengang: Informationstechnik und Elektrotechnik
Seminargruppe: MK05w3

Erstprüfer: Prof. Dr.-Ing. habil. Lutz Winkler
Zweitprüfer: Dipl.-Ing. Rico Thomanek

Mittweida, Mai 2010

Bibliografische Angaben

Robert Lukaschek:

„Konzeption und Entwicklung eines Personalinformationssystems“, Hochschule Mittweida (FH) – University of Applied Science, Fachbereich Informationstechnik und Elektrotechnik, Seminargruppe MK05w3, Diplomarbeit 2010

108 Seiten, 48 Literaturquellen, 29 Abbildungen, 9 Anlagen

Referat:

Vielen Unternehmen wird mehr und mehr bewusst, wie wichtig die Optimierung der Personalressourcen innerhalb der eigenen Firma ist. Dieser zentrale Aspekt der Personalverwaltung innerhalb eines Unternehmens erfordert Möglichkeiten, die firmeneigenen Daten effizient zu bündeln, zu verarbeiten und auszuwerten.

Die Fülle von Informationen, die heute für eine systematische Personalarbeit notwendig sind, macht computergestützte Personalinformationssysteme erforderlich.

Die Skriptsprachen Flash/Actionscript, PHP und MySQL bieten für die Erstellung eines Personalinformationssystems alle notwendigen Entwicklungswerkzeuge.

Ziel dieser Arbeit ist es, auf Basis von Flash, PHP und MySQL, entsprechende Entwicklungsvoraussetzungen für ein solches Projekt zu analysieren, die Struktur notwendiger Anwendungsbereiche zu erstellen und wichtige programmiertechnische Zusammenhänge zu erläutern. Der Leser sollte sich anhand dieser Arbeit allgemeine Grundkenntnisse zur Erstellung eines eigenen betriebsspezifischen Personalinformationssystems aneignen können.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	i
Vorwort.....	iv
Abbildungsverzeichnis.....	v
Abkürzungsverzeichnis.....	vii
0 Einleitung.....	1
0.1 Motivation.....	1
0.2 Kapitelübersicht.....	2
1 Personalwesen und Personalinformationssysteme.....	3
1.1 Personalwesen.....	4
1.2 Personalinformationssysteme.....	6
1.2.1 Definition eines Personalinformationssystems.....	6
1.2.2 Aufgaben und Ziele.....	6
1.3 Informationserfassung.....	8
1.3.1 Informationsart.....	8
1.3.2 Erfassung fixer Mitarbeiterinformationen.....	9
1.3.3 Dispositive Personalplanung mittels Personalinformationssystem.....	10
2 Projektmanagement.....	13
2.1 Vorgehensmodelle.....	13
2.1.1 Das klassische sequenzielle Phasenmodell.....	14
2.1.2 Das Wasserfall-Modell.....	15
2.1.3 Das V-Modell.....	16
2.1.4 Das Spiralmodell.....	16
2.1.5 Das Modell des Prototypings.....	17
2.1.6 Andere Vorgehensmodelle.....	18
2.2 Spezifikationsstufen eines Softwaresystems.....	18
2.2.1 Die Architekturbasis.....	18
2.2.2 Die Makroarchitektur.....	19
2.2.3 Die Mikroarchitektur.....	20
2.2.4 Die Technische Architektur.....	21
2.3 Anwendung des Phasenmodells auf den Entwicklungsprozess eines Personalinformationssystems in dieser Arbeit.....	21
3 Analyse der Rahmenbedingungen.....	23
3.1 Die Hardwarekomponenten und die Softwarearchitektur.....	23
3.1.1 Die Hardwarekomponenten.....	23

3.1.2	Softwarearchitektur	25
3.2	Besondere Programmanforderungen durch den Benutzer	29
3.2.1	Sicherheit	30
3.2.2	CMS - Weiterentwicklung	30
3.2.3	Benutzergruppen.....	32
3.3	Verwendete Technologien.....	33
3.3.1	MySQL zur Speicherung von Daten	34
3.3.2	PHP5 für die Anwendungs- und Persistenzschicht.....	37
3.3.3	Flash CS4 für die grafische Benutzeroberfläche.....	38
4	Konzeption eines Personalinformationssystems.....	43
4.1	Das Grobkonzept der Beispielanwendung.....	43
4.2	Konzeption des Login-Bereichs mit Preloading-Prozess	45
4.2.1	Der Login-Bereich	45
4.2.2	Der Preloading-Prozess	46
4.3	Konzeption des Personalbereichs	47
4.3.1	Darstellung der Personalinformationen	47
4.3.2	Darstellung der Personalkompetenzen.....	48
4.3.3	Verwaltung der Personalinformationen und -kompetenzen	49
4.4	Konzeption der Steuerprozesse	50
4.4.1	Die Personalverteilung (Dienstplanerstellung).....	51
4.4.2	Raumplanung.....	54
4.5	Konzeption des Kalenderbereichs.....	56
4.5.1	Kalenderübersicht	56
4.5.2	Kalenderinformationen	58
4.6	Konzeption des Operatorbereiches	60
4.6.1	Aufgaben eines Operators	60
4.6.2	Hintergrundinformationen und Runs.....	60
4.6.3	Erstellung und Bearbeitung der Hintergrundinformationen	61
4.6.4	Die Runbesetzung.....	62
4.7	Fazit der Konzeption	63
5	Realisierung eines Personalinformationssystems.....	65
5.1	Die Datenbank- und Ordnerstruktur.....	65
5.1.1	Datenbankstruktur der Beispielanwendung	65
5.1.2	Die Ordnerstruktur.....	66
5.2	Realisierung des Anmeldebereichs	67
5.2.1	Die Schnittstelle Actionscript/PHP (URLRequest, URLLoader, URLVariables)....	68
5.2.2	PDO (PHP Data Objects) als Datenbankschnittstelle	70

5.2.3	Das Verwalten der Benutzerrechte.....	70
5.2.4	Der Direktzugriff auf nichtauthentifizierte Programmdateien.....	72
5.3	Grundstruktur der grafischen Benutzeroberfläche	73
5.3.1	Die Movie-Clip Klasse	74
5.3.2	Die Menükomponente	75
5.3.3	Die Contentbox	76
5.3.4	Weitere Programmmodule	76
5.3.5	Das Einbinden externer SWF-Dateien.....	77
5.3.6	Grundgedanken zur Erstellung der Programmkomponenten.....	78
5.4	Das Initialisieren von Programmdateien und -inhalten.....	85
5.4.1	Preloading vs. Livestream	85
5.4.2	Der Preloading-Prozess	85
5.5	Die Umsetzung des Personalbereichs.....	88
5.5.1	Die Datenbanktabellen des Personalbereichs.....	89
5.5.2	Strukturierung der Datenbankabfragen	90
5.5.3	Der Umgang mit Bilddateien.....	91
5.5.4	Stander Eingabekomponenten der Informationsverwaltung	93
5.6	Die Implementierung von Steuerprozessen.....	95
5.6.1	Die Tabellenstruktur der Steuerprozesse	96
5.6.2	Manuelles Zuteilen von Personen	97
5.6.3	Automatische Verteilung mit Initialisierungsschlüssel.....	99
5.7	Realisierung der Kalenderfunktion.....	101
5.7.1	Die Tabellensätze der Kalenderübersicht.....	101
5.7.2	Die Auswahlkomponente für die Datumseinstellung.....	102
5.7.3	Das Bestimmen von Feiertagen	103
5.8	Realisierung des Operatorbereiches	104
5.8.1	Das Drucken ausgewählter Bildschirmkomponenten.....	104
5.8.2	Daten exportieren.....	105
5.9	Das Abmelden eines Benutzers	106
6	Ergebnisse und Ausblick.....	107
6.1	Fazit.....	107
6.2	Ausblick.....	107
	Anlagen.....	109
	Literaturverzeichnis	115
	Erklärung.....	123

Vorwort

Hiermit möchte ich mich besonders bei Prof. Dr.-Ing. habil. Lutz Winkler für die gute Unterstützung und Betreuung bei dieser Arbeit bedanken. Das selbige gilt für Dipl.-Ing. Rico Thomaneck, durch dessen Anteilnahme diese Arbeit überhaupt erst ermöglicht wurde.

Ein ganz großes Dankeschön geht an meine Familie, die mir nicht nur in zahlreichen Stunden dieser Arbeit, sondern auch in vielen wichtigen Situationen der letzten Jahre unterstützend zur Seite stand.

Für die zahlreichen Ideen, Hinweise und konstruktive Kritik bedanke ich mich insbesondere bei meinen ehemaligen Kommilitonen Ralf Sichert, Sven Porkert und Thomas Emmerich – ohne Euch hätte ich es nicht bis zum Diplom geschafft.

Weiterer Dank geht an Claudia, Kamal, Konni, Stefanie, Frank und Diana fürs Motivieren und „Dran-Denken“.

Abbildungsverzeichnis

<i>Abbildung 1: Aufgaben des Personalwesens.....</i>	<i>4</i>
<i>Abbildung 2: Hauptaufgaben eines PIS.....</i>	<i>7</i>
<i>Abbildung 3: Entwicklungsstufen und Ablaufplan des Phasenmodells.....</i>	<i>14</i>
<i>Abbildung 4: Komponenten- und Konzeptdarstellung des Wasserfall-Modells.....</i>	<i>15</i>
<i>Abbildung 5: Komponenten- und Konzeptdarstellung des V-Modells.....</i>	<i>16</i>
<i>Abbildung 6: Konzeptdarstellung des Spiralmodells</i>	<i>17</i>
<i>Abbildung 7: Model-View-Control (Schema).....</i>	<i>20</i>
<i>Abbildung 8: Phasen des Entwicklungsprozesses eines PIS in dieser Arbeit</i>	<i>22</i>
<i>Abbildung 9: Fat-Client-Architektur.....</i>	<i>27</i>
<i>Abbildung 10: Thin-Client-Architektur</i>	<i>28</i>
<i>Abbildung 11: Web-Server-Architektur</i>	<i>29</i>
<i>Abbildung 12: Zu erfüllende Kriterien von MySQL in Bezug auf die Beispielanwendung</i>	<i>34</i>
<i>Abbildung 13: Grobkonzept der Beispielanwendung</i>	<i>44</i>
<i>Abbildung 14: UML des Anmelde- und Preloading-Prozesses.....</i>	<i>46</i>
<i>Abbildung 15: Entwurfsmuster Personalkompetenzen.....</i>	<i>48</i>
<i>Abbildung 16: Entwurfsmuster Steuerprozesse - Personen verteilen.....</i>	<i>54</i>
<i>Abbildung 17: Entwurfsmuster für die Raumverteilung</i>	<i>55</i>
<i>Abbildung 18: Entwurfsmuster für die Kalenderübersicht eines Teamleiters.....</i>	<i>57</i>
<i>Abbildung 19: Entwurfsmuster der Kalenderinformation eines Teamleiters</i>	<i>59</i>
<i>Abbildung 20: Entwurfsmuster der Runverwaltung eines Teamleiters bzw. Admins</i>	<i>61</i>
<i>Abbildung 21: Beispielvariante der Ordnerstruktur eines PIS</i>	<i>66</i>
<i>Abbildung 22: Datenbanktabellen des Anmeldebereichs.....</i>	<i>71</i>
<i>Abbildung 23: Anordnung der MovieClips.....</i>	<i>74</i>
<i>Abbildung 24: Tabellensatz der Personalinformation.....</i>	<i>90</i>
<i>Abbildung 25: Häufig genutzte Editierkomponenten in einem Personalinformationssystem..</i>	<i>94</i>
<i>Abbildung 26: Tabellenauszug einer möglichen Dienstplanverteilung der Mitarbeiter.....</i>	<i>96</i>
<i>Abbildung 27: Grundprinzips der automatischen Dienstplanerstellung</i>	<i>99</i>
<i>Abbildung 28: Tabellensatz des Kalenderbereichs</i>	<i>102</i>
<i>Abbildung 29: Beispielvariante einer Datumskomponente</i>	<i>102</i>

Abkürzungsverzeichnis

AS	Dateiformat zum Ausführen von Actionscript
BLOB	Binary Large Object
CMS	Content Management System
CSV	Comma-Separated Values
FLV	Flash Video
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ID	Identifikationsschlüssel zur eindeutigen Zuordnung eines Datensatzes in der Datenbank
MyISAM	My Indexed Sequential Access Method
MySQL	Relationales Datenbankverwaltungssystem
PDA	Personal Digital Assistant
PDF	Adobe Portable Document Format
PIS	Personalinformationssystem
PHP	Hypertext Preprocessor
Run	bezeichnen in dieser Arbeit Zyklen zur Informationsbestimmung mit einer eigenen Zeitvorgabe
SWF	Shockwave Flash
UML	Unified Modeling Language
URL	Uniform Resource Locator
XAMPP	Kombinierter Einsatz von beliebigen Betriebssystem, Apache, MySQL und PHP bzw. PEARL
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

0 Einleitung

0.1 Motivation

„Es sind die Menschen, die über den zukünftigen Erfolg oder Misserfolg eines Unternehmens entscheiden. Das Personal ist heute weniger ein unliebsamer Kostenfaktor, als vielmehr das Leistungspotenzial eines Unternehmens.“

Vielen Unternehmen wird mehr und mehr bewusst, wie wichtig die Optimierung der Personalressourcen innerhalb der eigenen Firma ist. Nicht nur im Bezug auf Neueinstellungen, sondern stärker noch in der Ausschöpfung und Erweiterung der Potenziale der eingestellten Mitarbeiter, gibt es Raum zur Kosteneinsparung durch eine höhere Mitarbeitereffizienz. Dieser zentrale Aspekt der Personalverwaltung innerhalb eines Unternehmens erfordert Möglichkeiten, die firmeneigenen Daten effizient zu bündeln, zu verarbeiten und auszuwerten.

Die Fülle von Informationen, die heute für eine systematische Personalarbeit notwendig sind, macht computergestützte Personalinformationssysteme erforderlich. Diese stellen die Datenbasis für den personalwirtschaftlichen Aufgabenkomplex zur Verfügung. Folglich greifen immer mehr Unternehmen, neben zum allgemeinen Gebrauch entwickelten Softwareprodukten, auf speziell für den eigenen Betrieb entworfene Anwendungen zu. Die Fachkompetenzen des Personals, die unternehmensinterne Raumplanungen oder eine auf den Betrieb zugeschnittene Informationsverarbeitung können optimal erfasst und ausgewertet werden. Die Konsequenz ist eine enorme Steigerung des Firmenpotenzials.

Die Skriptsprachen Flash/Actionscript, Php und MySQL bieten für die Erstellung eines Personeninformationssystems alle nötigen Entwicklungswerkzeuge. Neben für grafische Tools benötigte APIs seitens Flash/Actionscript, der serverseitigen Schnittstellenbeschreibung von Flash auf MySQL durch das PHP-Script und einem sehr guten Datenbankhandling mittels MySQL werden alle Entwicklungsbereiche gut unterstützt.

Ziel dieser Arbeit ist es, auf Basis von Flash, PHP und MySQL, entsprechende Entwicklungsvoraussetzungen für ein solches Projekt zu analysieren, die Struktur notwendiger Anwendungsbereiche zu erstellen und wichtige programmiertechnische Zusammenhänge zu erläutern. Der Leser sollte sich anhand dieser Arbeit allgemeine Grundkenntnisse zur Erstellung eines eigenen betriebsspezifischen Personalinformationssystems aneignen können.

0.2 Kapitelübersicht

Informationen über das Personalwesen und Personeninformationssystemen bildet das Fundament dieser Diplomarbeit. Ohne grundlegende Zusammenhänge wird es schwer, verschiedene Begriffe zu verstehen. Mit **Kapitel 1 – Personalwesen und Personalinformationssysteme** soll dem Leser ein Basiswissen vermittelt werden.

In **Kapitel 2 - Projektmanagement** werden Grundzüge der Projektplanung erfasst. Es werden verschiedene Vorgehensmodelle zum Planen eines Softwareentwicklungsprozesses beschrieben. Exemplarisch werden die einzelnen Schritte und die allgemeiner Anordnung dieser auf den Entwicklungsprozess eines Personalinformationssystems projiziert. Die dafür gewählten Phasen repräsentieren gleichzeitig die Anordnung darauffolgender Kapitel in dieser Arbeit.

Das **Kapitel 3 – Analyse der Rahmenbedingungen** analysiert konzeptions- und entwicklungsvorbereitende Maßnahmen. Es werden neben den hardwaretechnische Voraussetzung, den verschiedenen Nutzeraspekten und der zu wählenden Softwarearchitektur insbesondere die verwendeten Technologien Flash/Actionscript, PHP und MySQL diskutiert.

Mit der Analyse und Konzeption der Programmstruktur soll sich **Kapitel 4 – Konzeption eines Personalinformationssystems** beschäftigen. Anhand einer Beispielanwendung werden die wichtigsten Kernbereiche eines Personalinformationssystems dargestellt.

Das **Kapitel 5 – Realisierung eines Personalinformationssystem** erforscht angelegt an dem Kapitel der Konzeption verschiedene bei der Entwicklung zu beachtende Realisierungsaspekte. Die Umsetzung der Beispielanwendung steht dabei nicht im Vordergrund. Vielmehr werden auftretende Probleme bzw. grundsätzliche Herangehensweisen bei der Erstellung eines Personalinformationssystems erfasst und anhand einzelner Quellcodebeispiele in allgemeiner Form beschrieben.

Abschließend wird in **Kapitel 6 – Ergebnisse und Ausblick** ein Resümee über den Gesamterlauf dieser Arbeit gezogen und zukünftige Entwicklungsaspekte analysiert.

1 Personalwesen und Personalinformationssysteme

In der Wirtschaft, der Forschung und der Lehre wird der Faktor Mensch zunehmend als erfolgsentscheidendes Kriterium für die Wettbewerbsfähigkeit eines Unternehmens gesehen¹. Das effektive Einsetzen der im Betrieb angestellten Mitarbeiter bewirkt neben Kosteneinsparungen auch eine produktivere, fehlerfreiere und kreativere Arbeitsweise der einzelnen Personen². Die Personalwirtschaft definiert dabei den zuständigen Bereich eines Unternehmens, welcher sich mit den arbeitenden Menschen und den durch diese verursachten personellen und sozialen Aufgaben beschäftigt³. Laut Jung berücksichtigt diese Definition sowohl die

Unternehmensbedürfnisse, welche in der bestmöglichen Versorgung des Unternehmens mit geeigneten Mitarbeitern besteht, und die

Mitarbeiterbedürfnisse, da für die Mitarbeiter eines Unternehmens zu sorgen ist. Die Mitarbeiter müssen betreut, entwickelt, verwaltet und entlohnt werden⁴.

Beide Bereiche bilden wichtige Kostenfaktoren innerhalb eines Unternehmens. Im Jahr 2004 hat eine geleistete Arbeitsstunde in Deutschland im produzierenden Gewerbe und im Dienstleistungsbereich durchschnittliche Arbeitskosten in Höhe von 28,18 Euro verursacht. Ein Vollzeitarbeitsplatz kostete den Arbeitgeber 47.129 Euro im Jahr⁵. Gerade die jährlich steigenden Personalkosten⁶ verursachen einen großen Teil dieser Ausgaben. Nur die optimale Nutzung der Unternehmens- und Mitarbeiterkapazitäten kann das Unternehmen auf Dauer wettbewerbsfähig halten. Ein zielloser Personaleinsatz, nicht besetzte Arbeitsplätze oder ungenutztes Entwicklungspotenzial von Mitarbeitern würden eine Ressourcenverschwendung bedeuten und dem Unternehmen auf Dauer schaden.

Der für Analyse, Umsetzung und Kontrolle verantwortliche und damit der wichtigste Bereich der Personalwirtschaft ist das Personalwesen.

¹ Vgl. [HoeselbSchulz2005], 200

² Vgl. [Jung2006], 891

³ Vgl. [Scholz2008], 16

⁴ Vgl. [Jung2008], 4

⁵ Vgl. [StaBundesamtD2004] Statistisches Bundesamt Deutschland 2004, „Was kostet Arbeit in Deutschland“, 7

⁶ Statistisches Bundesamt Deutschland 2004: „In der langfristigen Betrachtung sind die Personalkosten je geleistete Arbeitsstunde in Deutschland vom 1. Quartal 1996 bis zum 4. Quartal 2004 saisonbereinigt [...] um 19,8% gestiegen.“

1.1 Personalwesen

Das Personalwesen beschreibt die Organisationseinheit der Personalwirtschaft⁷. Alle Vorgänge und Maßnahmen, die sich mit den Angestellten innerhalb eines Betriebes von der Einstellung über den Einsatz und die Betreuung bis zur Beendigung des Arbeitsverhältnisses befassen, werden von Mitarbeitern der Personalabteilung und den zuständigen Abteilungsleitern bearbeitet und kontrolliert. Die wichtigsten Aufgaben lassen sich durch folgende Teilbereiche beschreiben:



Abbildung 1: Aufgaben des Personalwesens⁸

Prognose des Personalbedarfs – Die Einschätzung der sich im Betrieb befindlichen Arbeitsplatzressourcen und der daraus resultierenden Anzahl benötigter Mitarbeiter bilden die Basis und damit eine der wichtigsten Aufgaben⁹ für die Arbeit einer Personalabteilung.

Maßnahmen bei Personalbedarf – Durch nicht besetzte Arbeitsplätze gehen einem Unternehmen wichtige Produktionsquellen verloren¹⁰. Mittels interner Stellenverschiebungen kann dieser Bedarf gedeckt werden. Ist das nicht möglich, muss der Betrieb zwangsweise neue Mitarbeiter einstellen. Die Maßnahmen zur Gewinnung von Mitarbeitern in den erforderlichen Qualifikationen bilden einen wichtigen Aufgabenbereich des Personalwesens.

Mitarbeiterkompetenzen ausschöpfen – Die falsche Zuordnung von Arbeitsplätzen an nicht geeignete Arbeitskräfte führt zu einer mangelhaften Ausnutzung des „Humankapitals“¹¹. Das Personalwesen hat die Aufgabe herauszufinden, wie Mitarbeiter im Hinblick auf die Anforderungen des Arbeitsplatzes und unter Berücksichtigung ihrer persönlichen Fähigkeiten optimal eingesetzt werden können.

Personalförderung und -ausbildung – Nutzen Betriebe die Wertschöpfungspotenziale ihrer Angestellten nicht vollständig, zum Beispiel verursacht durch geringe Motivation oder

⁷ Vgl. [Jung2006], 893

⁸ eigene Grafik, Vgl. [Jung2006], 893

⁹ Vgl. [DrummHJ2008], 231

¹⁰ Vgl. [BundWirtTechn2007], 25

¹¹ Begriff „Humankapital“ nach [Kamaras2003], 12

durch nicht erkanntes Entwicklungspotenzial des Mitarbeiters, kommt es zur Ressourcenverschwendung innerhalb des Unternehmens. Nur in enger Zusammenarbeit mit den Angestellten kann das Personalwesen erkennen, welche Probleme und welche Talente sich in den einzelnen Arbeitsbereichen verstecken. Ziel des Personalwesens ist, diese zusätzlichen Humanressourcen zu entdecken und zu Wettbewerbsvorteilen des Unternehmens weiterzuentwickeln¹².

Kontrolle des Personals und deren Kosten – Die Personalabteilung oder mehr noch die einzelnen Abteilungsleiter sind in der Pflicht zu kontrollieren, ob ein angestellter Mitarbeiter die ihm zugeteilten Aufgaben ordnungsgemäß durchgeführt hat. Kosten, welche aus den geplanten personellen Maßnahmen entstehen, müssen gleichzeitig überwacht und im Sinne des Unternehmens gesteuert werden.

Maßnahmen bei Personalüberhang – Hat ein Betrieb zu viele Mitarbeiter kommt es in den wenigsten Fällen zu sofortigen betriebsbedingten Kündigungen. Personalumbau oder ein Einstellungsstopp mit der Konsequenz des Mitarbeiterabbaus durch Rentenabgänge können Lösungen für einen Personalabbau sein. Ziel ist es herauszufinden, wie überzählige Mitarbeiter unter weitgehender Vermeidung von sozialer Härte abgebaut werden können.

Durch eine gut aufgestellte Personalabteilung wird ein geregelter Betriebsablauf gewährleistet und das Entstehen unnötiger Kosten durch Ressourcenverschwendung vermieden. Laut einer deutschlandweiten Studie der Universität Köln betreut ein Angestellter des Personalwesens durchschnittlich 88 Mitarbeiter¹³ des jeweiligen Unternehmens. Mit steigender Größe der Belegschaft verschlechtert sich diese Quote. Je mehr Angestellte in einem Unternehmen tätig sind, umso ineffektiver arbeitet eine Personalabteilung¹⁴. Um Kosten zu sparen, lassen viele Unternehmen ihr Personalwesen unterbesetzt, was dieses bezüglich des verwaltungstechnischen Aufwands an die Belastungsgrenzen führt.

Eine alternative Lösung zur Minderung der Personalkosten bei gleichzeitiger Entlastung des Personalwesens stellen sogenannte Personalinformationssysteme dar.

¹² Vgl. [Brauweiler2008], 177

¹³ Vgl. Studie [HRAmpeICheck32007]“, 23

¹⁴ Vgl. [StatBundAmt2004] Statistisches Bundesamt 2004: „Was kostet Arbeit“, 15

1.2 Personalinformationssysteme

Ein Personalinformationssystem ermöglicht die Erfassung, Speicherung, Verarbeitung, Pflege, Analyse, Benutzung, Verbreitung, Disposition, Übertragung und Ausgabe von Informationen zur Unterstützung administrativer und dispositiver personalwirtschaftlicher Aufgaben¹⁵. Betriebsinterne Prozesse können wesentlich schneller, aktueller und umfassender bewältigt werden. Die in dieser Arbeit abgekürzt als PIS bezeichneten Systeme unterstützen die Aufbereitung von Datenmaterial und dienen als umfassende Informationsgrundlage bzw. Entscheidungshilfe für zukünftige Problembereiche der Personalplanung und -steuerung.

1.2.1 Definition eines Personalinformationssystems

Ein Personalinformationssystem, auch Human Resource Information System¹⁶ genannt, wird definiert als ein System zur geordneten Erfassung, Speicherung, Transformation und Ausgabe aller relevanten Informationen der in einem Betrieb vorhandenen Personal- und Arbeitsbereiche. Wobei diese Informationen an berechnigte Führungskräfte, Arbeitnehmer und Ihre Vertretungen ausgegeben werden, damit diese ihre Aufgaben wahrnehmen können. Darüber hinaus kann das Personalinformationssystem zur Versorgung mit Informationen weiterer Interessensgruppen, wie der Öffentlichkeit, Betriebsräten oder Behörden, genutzt werden¹⁷.

1.2.2 Aufgaben und Ziele

Die Aufgaben eines Personalinformationssystems beziehen sich auf die Bereitstellung von Informationen zu verschiedenen Zwecken. In welchem Ausmaß ein Personalinformationssystem für betriebliche Prozesse unterstützend eingesetzt wird, ist vom jeweiligen Unternehmen abhängig. Eine genaue Erörterung und Ausgrenzung von Anwendungsbereichen des Systems spielt eine wichtige Rolle und führt neben der einfacheren und schnelleren Handhabung der Software und dem effektiveren Einarbeiten von Mitarbeitern auch zu kürzeren Entwicklungszeiten und den damit verbundenen Kosten.

¹⁵ Vgl. [Muelder2000], 98

¹⁶ Vgl. [Oechsler2005], 193

¹⁷ Vgl. [GruenefLangem1991], 18

Die wichtigsten Aufgaben eines PIS sind zu informieren, zu steuern und zu kontrollieren¹⁸.



Abbildung 2: Hauptaufgaben eines PIS¹⁹

Die Aufgaben des **Informierens** beziehen sich auf die Bereit- und Darstellung von firmenabhängigen Daten für bestimmte Nutzergruppen des Unternehmens. Auf Kontaktdaten und Mitarbeiterinformationen kann schneller zugegriffen werden, Bereiche der Unternehmensplanung werden unterstützt und Konfliktsituationen zwecks Einsatzplanung werden frühzeitig aufgedeckt. Allein das schnelle Abrufen und die übersichtliche Wiedergabe der für betriebliche Prozesse notwendigen Daten steht bei der Methode des Informierens im Vordergrund. Oft werden Suchfunktionen und an das Unternehmen angepasste Verzeichnisstrukturen zu diesem Zweck eingesetzt. Anwendungsbereiche für die Methode des Informierens sind: Personalstammdatenverwaltung, Darstellung interner Personalstatistiken, Einsatzplanung und Statistiken oder Stellenanforderung eines Arbeitsplatzes.

Werden unternehmensbezogene Informationen zur Führung betriebsinterner Abläufe genutzt, so werden diese mit der Methode **Steuern** bezeichnet. Gerade bei den dispositiven Prozessen werden Personalinformationssysteme auf Grund der Vielzahl an Einzelinformationen²⁰ in den verschiedensten Bereichen unterstützend eingesetzt. Steuern bedeutet vorhanden Informationen auszuwerten, Arbeitsprozesse zu koordinieren und anhand dieser, neugewonnene Informationen für weitere Aufgaben zu sichern. Im Wesentlichen konzentrieren sich die Funktionen des Steuerns auf die Personalbedarfsermittlung, Personaleinsatzplanung, Personalbeschaffung, Personalentwicklung, Personalerhaltung und Personalfreistellung.

Die Methode Kontrollieren dient dem Nachweis der ordnungsgemäßen Erfüllung festgelegter Aufgaben. Mit Hilfe aufbereiteter Daten lassen sich im Unternehmen differenzierte Soll/Ist-

¹⁸ Vgl. [Schroeder2009], 10 ff.

¹⁹ Vgl. [Schroeder2009], 11

²⁰ Vgl. [Oechsler2005], 193

Vergleiche durchführen²¹, die neben den reinen Zahlenwerten auch Ursachen für Abweichungen aufzeigen. Betriebsräte und Führungsetagen können mit Hilfe dieser Daten jederzeit einen planungsmäßigen Ablauf betriebsinterner Handlungen überprüfen. Beispiele für die Methode "Kontrollieren" sind unter anderem: die Einhaltung der Personalplanung, Arbeits- und Ausfallzeitanalysen, Personalaufwandsanalysen, Anwesenheitskontrollen, die Fehlzeitenverwaltung und -auswertung und die Terminüberwachung.

1.3 Informationserfassung

Die Voraussetzung für den nützlichen Einsatz eines Personalinformationssystems bildet die Erhebung einer auf den Betrieb zugeschnittenen Datenbasis. Jedes Unternehmen muss selbst festlegen, welche Informationen für anstehende Arbeitsabläufe relevant sind und in welchem Maße Mitarbeitern entsprechende Kompetenzen zugesprochen werden.

1.3.1 Informationsart

Die im Unternehmen vorherrschenden Arbeitsprozesse bilden die Grundlage für die Bewertung und Einsatzplanung der Mitarbeiter. Aus ihnen entstehen speziell auf den Betrieb zugeschnittene Informationsinhalte, mit deren Hilfe Kompetenzen eines Angestellten bestimmt werden können. Die wichtigsten Informationsarten in einem Personalinformationssystem werden durch drei Sichtweisen beschrieben: signifikante Angaben des Mitarbeiters oder des Arbeitsplatzes, zyklisch-temporäre Informationen und Daten im Sinne langfristiger Planung.

Signifikante Informationen sind fest an eine Person oder einen Arbeitsplatz gebundene und in den meisten Fällen unveränderbare Daten. Sie sind nicht von dem Urteil anderer Personen abhängig und ergeben sich ausschließlich durch vorherrschende feststehende Zustände. Gegenstand solcher Informationen sind beispielweise die Kontaktdaten der Mitarbeiter, die obligatorische Raumplanung oder die Mitarbeiterhierarchie im Unternehmen. Diese Angaben dienen rein der Methode des Informierens. Personalinformationssysteme haben für diese Art der Information verstärkt auf die zweckmäßige Anordnung von Inhalten, das heißt wichtige Informationen stehen im Vordergrund, und auf möglichst kurze Zugriffszeiten zu achten.

²¹ Vgl. [Schroeder2009], 10

Zyklisch-temporäre Informationen sind Daten des täglichen Gebrauchs und dienen der Planung kurzzeitiger Unternehmensabläufe sowie der Erschließung von Mitarbeiterkompetenzen. In welcher Form diese Werte Bestand haben, hängt vom jeweiligen Unternehmen ab. Anhand der Struktur des Betriebes und der vorherrschenden Arbeitsabläufe ergibt sich in den meisten Firmen ein vorherrschendes Planungsmuster. Benötigen mehrere Mitarbeiter unabhängig ihrer Kenntnisse die gleichen Arbeitsplatzressourcen, so steht die Raumplanung im Vordergrund. Werden zu festgelegten Zeiten spezifische Fachkenntnisse gefordert ist die Personalverteilung entscheidend. Dieses Planungsmusters gilt es zu erkennen und für die Entwicklung des PIS durch die Bezeichnung aller vorhandenen Kriterien genau zu beschreiben. Wird jedes Argument zusätzlich durch eigene Unterkriterien definiert, ist neben der Einsatzplanung auch die Beurteilung von Mitarbeitern in verschiedenen Fachgebieten möglich. Leitende Personen können mit Hilfe dieser Informationen für einen ausgeglichenen Personaleinsatz sorgen, den bestgeeignetsten Mitarbeiter situationsbedingt einsetzen oder Mangel von Personalkompetenzen frühzeitig erkennen.

Eine dritte Variante der Information sind Daten im Sinne langfristiger Planung. So führen Urlaub, Weiterbildungen, Rentenabgänge oder Krankmeldungen zu betriebsinternen Ausfällen des Personals. Folglich ergibt sich eine Mitarbeiterunterbesetzung bezüglich der täglichen Einsatzplanung. Mit einem Personalinformationssystem können derartige Konfliktsituationen frühzeitig erkannt und durch eine manuelle oder automatisierte Umgestaltung des Personaleinsatzes rechtzeitig beseitigt werden. Längerfristige Daten werden mit Abstimmung des Vorgesetzten festgelegt und direkt in den Tagesbetrieb eingebunden.

1.3.2 Erfassung fixer Mitarbeiterinformationen

Die elementarsten Daten sind in den meisten Unternehmen personengebundene Mitarbeiterdaten. Die Aufbereitung dieser Daten ist an einem bestimmten Empfängerkreis auszurichten. Die sogenannte Personalberichterstattung, welche entweder durch die Personalabteilung, den Leiter eines Unternehmensbereichs oder den Mitarbeiter selbst vertreten ist, hat die Verantwortung alle notwendigen Daten zur weiteren Nutzung anzulegen. In den meisten Fällen geschieht dies durch vorgefertigte Personalbögen. Personeninformationssysteme wirken dabei unterstützend oder übernehmen in einem gewissen Rahmen die Arbeit der Personalberichterstattung. Ändert sich der Status einer Person ist dies unmittelbar dem Personalwesen mitzuteilen. Bei dieser Art der Information ist unbedingt zu klären, in welchem Maß und an welche Nutzergruppen die Daten veröffentlicht werden. Es existieren zu diesem

Zweck, sei es durch allgemeine Vorschriften bedingte oder durch den Betriebsrat festgelegte, streng einzuhaltende Datenschutzbestimmungen.

1.3.3 Dispositive Personalplanung mittels Personalinformationssystem

Durch immer komplexere Aufgaben und Problemstellungen innerhalb der Personalplanung ergibt sich ein wachsender Informations- und Entscheidungsbedarf im dispositiven Bereich²². Gerade die tägliche Einsatzplanung wird in vielen Fällen schlecht organisiert, wodurch nur ein Teil der vorhandenen Human Ressourcen eines Betriebes wirklich genutzt wird. Durch den Einsatz eines PIS können dispositive Prozesse rascher und flexibler erledigt werden. Die Voraussetzung für ein solches System ist die Verknüpfung von Arbeitsplatzinformationen und Mitarbeiterkompetenzen. Personalinformationssysteme können dabei rein informierend wirken oder komplette Steuerungsprozesse übernehmen. In welchem Maß ein PIS eingesetzt wird, ist vom Einsatzwillen des Unternehmens abhängig.

Bei halbautomatischen Systemen wird das Personalinformationssystem als Informationsapparat eingesetzt. Die Einsatzplanung wird mit Hilfe der bereitgestellten Informationen durch den Mitarbeiter selbst durchgeführt. Mit dem Ablauf von Arbeitsintervallen können Kompetenzbewertungen erkannt und diese durch den Vorarbeiter oder durch das System aktualisiert werden. Der eigenständige Handlungsbedarf seitens des Nutzers ist absolut notwendig. Durch das unterstützende PIS kann dieser im Vergleich zur analogen Arbeitsweise jedoch vereinfacht und effektiver gestaltet werden. Im Vergleich zu vollautomatisierten Systemen behält der Benutzer die absolute Transparenz aller vom System vorgenommenen Arbeitsabläufe. Gerade bei langfristigen Planungszielen, zum Beispiel der Urlaubsplanung finden halbautomatisierte Systeme ihren Nutzen. Mögliche Konfliktsituationen sollten von dem System sofort angezeigt, auf Wunsch geblockt und damit grundsätzlich verhindert werden. Bereits durch andere Mitarbeiter belegte Ressourcen, können als besetzt gekennzeichnet und von einer weiteren Verwendung ausgeschlossen werden. Auch Weiterbildungsmaßnahmen und Kranktage können unteren anderen auf diese Weise verwaltet werden.

Im Gegensatz dazu übernehmen vollautomatisierte Systeme die Aufgaben der Steuerung von betrieblichen Prozessen. Mitarbeitereigenschaften und Arbeitsplatzressourcen werden vom System erkannt, ausgewertet und aufeinander abgestimmt. Das System erkennt beendete Arbeitsprozesse und übernimmt automatische eine Neubewertung des eingesetzten

²² Vgl. [Jung2006], 711

Personals. Der Arbeitsaufwand gestaltet sich bei dieser Lösung am geringsten. Der Entwicklungsaufwand ist hier allerdings um einiges größer als bei halbautomatisierten Systemen. Diese Art der Anwendung wird oft bei Prozessen mit kurzen Planungsintervallen zum Beispiel bei der täglichen Einsatzplanung benutzt.

Die Kombination beider Systemtypen stellt eine optimale Lösung zur Nutzung eines Personalinformationssystems dar. Vom System angelegte Prozesse sollten bei Bedarf individuell durch den Nutzer verändert und fehlerhafte durch die Anwendung angelegte Prozesse jederzeit korrigiert werden können.

2 Projektmanagement

Wird Software geschäftsprozessunterstützend eingesetzt, dann gilt es zunächst zu entscheiden, ob die Anwendung selbst entwickelt oder bereits vorhandene Softwareprodukte genutzt werden sollen. Sowohl die Beschaffung als auch die Entwicklung von Personalinformationssystemen führen zu einer Vielzahl unterschiedlich zu koordinierenden Aufgaben, bei denen in der Regel mehrere Personen beteiligt sind. Erfahrungsgemäß scheitern Softwareprojekte häufig wegen ungelösten organisatorischen Problemen und seltener wegen nicht zu bewältigenden technischen Anforderungen. Einfach beginnen und denken: "Es wird schon laufen!" ist bei der Größe²³ vieler Projektes nicht mehr möglich. Einer konzeptionellen Projektplanung vor Start kommt daher eine besondere Bedeutung zu.

In den folgenden Kapiteln werden mögliche Vorgehensmodelle des Projektmanagements und die wichtigsten Schritte bei der Entwicklung eines Personalinformationssystems speziell im Hinblick auf den weiteren Verlauf dieser Arbeit beschrieben.

2.1 Vorgehensmodelle

Vorgehensmodelle, auch Prozessmodelle genannt, werden als wichtigste Hilfsmittel des Projektmanagements bezeichnet. Sie dienen der Benennung, Beschreibung und Ordnung von Tätigkeiten im Rahmen der Softwareentwicklung²⁴. Da jedes Projekt anders ist, gelingt es nur selten, ein Projekt genau nach einem bestimmten Modell zu konzipieren. Der Entwurf und die Entwicklung einer Anwendung benötigen in den meisten Fällen eine eigene projektbezogene Analyse möglicher Entwicklungsabläufe. Eine allgemeine Betrachtung der Vorgehensmodelle sollte trotzdem durchgeführt werden, da diese als theoretische Grundlage zur Erstellung eigener Prozessabläufe genutzt werden können.

Ausschlaggebend für die Auswahl von Projektführungsmethoden in dieser Arbeit sind der Verbreitungsgrad und die Praxisrelevanz dieser Modelle. Weniger bedeutende Vorgehensmodelle werden zwar genannt aber nicht detailliert beschrieben²⁵.

²³ Vgl. [GummSommer2006], 795: „Wenn die Programme einen Umfang von 1000 oder 2000 Zeilen übersteigen, versagt in der Regel ein völlig unsystematische Programmierung.“

²⁴ Vgl. [PombergerPree2004], 9

²⁵ Detaillierte Beschreibung dieser Vorgehensmodelle unter [Buhl2004], 16 ff.

2.1.1 Das klassische sequenzielle Phasenmodell

Ein bedeutender Ansatz im Projektmanagement ist das Konzept der Phasen. Software ist ab einer bestimmten Größe nicht in einem Zuge entwickelbar. Das Phasenmodell definiert, dass von der ersten Idee bis hin zum fertigen Produkt mehrere Entwicklungsstufen durchlaufen werden. Jede neue Entwicklungsstufe beginnt erst dann, wenn alle Aufgaben der vorhergehenden Phase abgeschlossen sind. Die Bestandteile eines solchen Modells sind im Wesentlichen durch die folgenden Phasen beschrieben:



Abbildung 3: Entwicklungsstufen und Ablaufplan des Phasenmodells

Diese streng sequenzielle Vorgangsweise bewirkt, dass Software-Projekte besser plan-, organisier- und kontrollierbar werden²⁶. Durch das klare Definieren und das voneinander Abgrenzen einzelner Tätigkeiten des Software-Entwicklungsprozesses wird ein eindeutiger und für alle Beteiligten nachvollziehbarer Projektplan erstellt. Die generierten Phasen ermöglichen das Ableiten von Meilensteinen, durch welche ein effektives Zeitmanagement bei der Entwicklung des Softwareprojekts realisiert werden kann. Die Strategie des Phasenmodells ist unabhängig von einem Anwendungsgebiet, von der Projektgröße und von der Komplexität des geplanten Produkts anwendbar. Desweiteren ist das Konzipieren arbeitsteiliger Entwicklungsprozesse möglich, welche für die Zusammenarbeit mehrerer Personen notwendig sind.

Der große Nachteil eines solchen Prozessmodells zeigt sich durch die (falsche) Annahme, dass der Entwicklungsprozess in der Regel sequentiell ausgeführt wird²⁷. Die Praxis zeigt, dass es nur in den seltensten Fällen im ersten Entwurf gelingt, eine vollständige Spezifikation oder ein fehlerfreies Konzept einer Systemarchitektur zu erstellen. Die nötigen Informationen für den vollständigen Abschluss einer Phase ergeben sich teilweise erst, durch später zu bearbeitende Phasen. Desweiteren ist es im klassischen Phasenmodell nicht zugelassen parallel phasenunabhängige Tätigkeiten abzuarbeiten. In Bezug auf die Entwicklungszeit und einer effektiven Arbeitsteilung werden gerade bei hohen Mitarbeiterkapazitäten eines Unternehmens schnell Ressourcen verschwendet.

²⁶ Vgl. [Taborsky2008], 9

²⁷ Vgl. [PombergerPree2004], 16

2.1.2 Das Wasserfall-Modell

Das Wasserfall-Modell, eine von Roice 1970 erweiterte und von Boehm seit 1973 verbreitete Variante des Phasenmodells, interpretiert die einzelnen Phasen als Tätigkeiten. Da das streng sequentielle Anordnen von Tätigkeiten nur bei den einfachsten Programmen möglich ist, sind im Wasserfall-Modell Zyklen vorgesehen, welche den ständigen Rückbezug auf im Phasenmodell bereits abgeschlossene Tätigkeiten ermöglichen²⁸. Um zu komplizierte Prozesse über mehrere Projektschichten zu vermeiden, wird nur der Bezug aufeinanderfolgender Phasen zugelassen. Das Konzept der Meilensteine ist in dieser Variante des Projektmanagements nicht realisierbar.

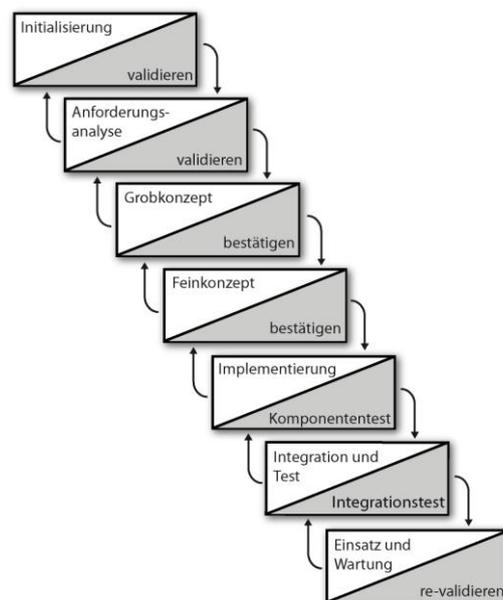


Abbildung 4: Komponenten- und Konzeptdarstellung des Wasserfall-Modells

Ergänzend können im Wasserfall-Modell Validierungsverfahren bezüglich der Phasenergebnisse in das Prozessmodell eingebunden werden, welche den Übergang in eine neue Phase zusätzlich absichern. Durch das mehrfache Überarbeiten der einzelnen Entwicklungsschichten, vor allem bei der Systemanalyse und der Konzeption, wird das Risiko von unvollständigen Systemspezifikationen und von Design-Fehlern vermindert. Fehlentscheidungen lassen sich so schneller erkennen und Software-Entwicklungsprozesse besser kontrollieren²⁹.

²⁸ Vgl. [FruehLudewigSandm2001], 14

²⁹ Vgl. [PombergerPree2004], 11

2.1.3 Das V-Modell

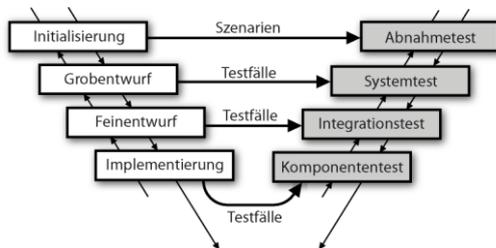


Abbildung 5: Komponenten- und Konzeptdarstellung des V-Modells

Das V-Modell ist eine Erweiterung des Wasserfall-Modells und wurde für den öffentlichen Bereich entwickelt. Es ist das genormte Standardvorgehensmodell für die Planung und Durchführung von Software-Projekten des Bundes, des Militärs und auch vieler Industrieunternehmen³⁰.

Das V-Modell gliedert den Entwicklungsprozess in eine genau definierte Sequenz von Phasen, welche wiederum durch IEEE-Standards, den ISO 12207³¹ und durch DIN/ISO 9001³² geprägt sind. Das V-Model XT³³, basierend auf dem 1997 definierten V-Model 97³⁴, ist der derzeit in Deutschland vorgeschriebene Standard für die Entwicklung von IT-Projekten.

Der Grundgedanke des Modells ist, für jede Schicht explizit den Qualitätssicherungsprozess zu integrieren und zwar so, dass jeder Phase des Qualitätssicherungsprozesses eine Phase des Entwicklungsprozesses zugeordnet ist. Dieses Vorgehen erleichtert den Projektbeteiligten die Umsetzung der Zielvorstellung und bietet in jeder Phase angemessene von der Entwicklung unabhängige Kontrollmechanismen für die Umsetzung.

2.1.4 Das Spiralmodell

Das Spiralmodell wird als iteratives Prozessmodell definiert. Kontrastiv zum sequentiellen Phasen-, Wasserfall- oder V-Modell werden die einzelnen Projektabschnitte dieses phasenorientierten Modells mehrfach zyklisch durchlaufen. Dabei entstehen mehrere Entwürfe bzw. vereinfachte Versionen (Prototypen) der zukünftigen Anwendung. Es wird in regelmäßigen Abständen geprüft, ob der jeweilige Prototyp weiterentwickelt werden muss oder zum Einsatz gelangen kann.³⁵ Der Verfeinerungsgrad der Entwicklung wird mit jedem Durchgang

³⁰ Vgl. [GadatschMayer2006], 234

³¹ Vgl. [TurnerPysterPennotti2009], 407 - 412

³² Vgl. [DINTaschenb2262009]

³³ Detaillierte Beschreibung unter: [vmodelXT2010]

³⁴ Detaillierte Beschreibung unter: [vmodel972010]

³⁵ Vgl. [BullingerFaehnrich1997], 12



Abbildung 6: Konzeptdarstellung des Spiralmodells

erhöht und kann direkt vom Kunden beeinflusst werden.

In Verbindung mit periodisch angelegten Risikoanalysen gibt das Spiralmodell zusätzlich eine kontinuierliche Übersicht, ob der Entwicklungsprozess den Anforderungen entspricht, dieser korrigiert oder sogar beendet werden muss³⁶.

Nachteile dieses Modells zeigen sich primär darin, dass die entsprechenden Risikoanalysen komplex und aufwendig sind. Der Kunde kann bei der Vorgehensweise des Spiralmodells Zwischenergebnisse schneller begutachten, der Entwicklungsprozess insgesamt dauert jedoch länger. Auch die entsprechende Projektsteuerung ist umfassender als in den Phasenmodellen. Der klare Ablauf geht gerade bei größeren Projekten schnell verloren und die Arbeit mit mehreren Entwicklungsgruppen gestaltet sich schwieriger.

2.1.5 Das Modell des Prototypings

Prototypingmodelle vertreten das Prinzip des iterativen Vorgehens noch intensiver als Spiralmodelle. Teilprozesse werden in enger Zusammenarbeit mit dem Kunden erstellt und nach ihrer Zielsetzung und ihrer funktionalen Ausrichtung unterschieden. Diese Art des Entwicklungsprozesses wird der Gruppe benutzermitwirkender Modelle zugeordnet³⁷. Projekterfolge sind schnell erkennbar, was sich wiederum positiv auf die Motivation des Projektteams und des Kunden während der Entwicklungszeit auswirkt.

Nachteile ergeben sich, durch das Verlieren des strukturierten Projektablaufs. Der ursprüngliche Gedanke eines Entwicklungsprozessmodells, das heißt einen geordneten Arbeitsprozess als Hilfsmittel der Projektführung zu erstellen, wird fast vollständig ignoriert.

³⁶ Vgl. [AbtsMuelder2008], 306

³⁷ Vgl. [BullingerFaehnrich1997], 13

2.1.6 Andere Vorgehensmodelle

Andere Modelle versuchen die sequentielle Projektbeschreibung zu umgehen. Ein sehr häufig verwendetes Prozessmodell dieser Methodik ist der „Rational Unified Process³⁸“. Entwicklungsprozesse dieser Art versuchen weg von der sequentiellen hin zur objektorientierten Prozessgestaltung zu gehen. Fast alle Prozesse können unabhängig voneinander bearbeitet, gekürzt oder durch neue Prozesse ergänzt werden. Der eigentliche Gedanke der Prozessgestaltung, einen übersichtlichen Projektablauf zu gestalten, wird nicht berücksichtigt. Aus diesem Grund wird der Einsatz dieser Vorgehensmodelle häufig in Verbindung mit einem der beschriebenen sequentiellen Prozessmodelle realisiert.

2.2 Spezifikationsstufen eines Softwaresystems

Die Spezifikationen, von einem ersten Entwurf bis hin zur Definition aller für die Implementierung notwendigen Parameter eines PIS, werden in verschiedenen Abstraktionsebenen durchgeführt. Diese unterscheiden sich durch ihren Detaillierungsgrad und können in der Regel in folgende Architekturmuster unterteilt werden:

- die Architekturbasis,
- die Makroarchitektur,
- die Mikroarchitektur und
- die technische Architektur.

2.2.1 Die Architekturbasis

Es existiert eine Reihe von Vorgaben und Festlegungen, welche die Struktur und fundamentale Mechanismen des PIS beeinflussen. Diese Kriterien sind vorerst nicht Bestandteil des Softwaresystems, können aber im späteren Verlauf der Konzeption und der Implementierung einen beachtlichen Einfluss auf den zu realisierende Entwicklungsaufwand erlangen. Es sollten unter anderem die letztendlichen Projektziele, der Einsatzbereich und die technischen Voraussetzungen für die Anwendung vor allen weiteren Entwicklungsphasen analysiert und formlos erfasst werden.

³⁸ Vgl. [Kruchten1999], 15

2.2.2 Die Makroarchitektur

Die Makroarchitektur beschreibt die grundlegende Struktur des Personalinformationssystems beschrieben. Die zu erstellenden Softwarekomponenten werden in einer kompakten Gesamtübersicht identifiziert und im Einzelnen detailliert skizziert. Die verwendeten Teilsysteme können bereits in der Konzeptionsphase verschiedenen Softwareschichten zugeteilt und notwendige Schnittstellen in Form der angebotenen Dienste spezifiziert werden.

Die Darstellung der Makroarchitektur erfolgt durch eigene Grafiken oder durch UML-Diagramme (Unified Modeling Language). Die Unified Modeling Language ist eine standardisierte Sprache für das Konzipieren von Software und anderen Systemen. Im Sinne einer Sprache definiert UML dabei Teilbereiche einer Software mit deren Bezeichnungen und legt mögliche Beziehungen zwischen diesen Begriffen fest.

Entwurfsmuster

Bei der Entwicklung eines Softwarekonzeptes helfen Entwurfsmuster (*design patterns*), welche bewährte Lösungsschablonen für wiederkehrende Entwurfsprobleme in der Softwareentwicklung beschreiben und ein grundlegendes Ordnungsprinzip zur Strukturierung eines Softwaresystems liefern. Sie helfen bei der Zerlegung des Systems in verschiedene Teilbereiche und veranschaulichen die Mechanismen, mit denen diese kommunizieren. Bezüglich des Gesamtkonstrukts einer Software wird die Sammlung einzelner Entwurfsmuster als Architekturmuster (*architecture pattern*) bezeichnet. Ein grundlegendes und für die Entwicklung eines PIS empfohlenes Architekturmodell basiert auf der Verwendung von Softwareschichten – das MVC-Modell (*model view control*).

Drei-Schicht-Architektur mit MVC

Die meisten Systeme sind aus drei Standard-Softwareschichten aufgebaut: der Präsentations-, der Anwendungs- und der Persistenzschicht³⁹. Diese drei Schichten reflektieren die fundamentalen Aufgaben von Softwaresystemen. Ein Personalinformationssystem benötigt in der Regel eine grafische Benutzeroberfläche, um Daten darzustellen und auf Benutzereingaben zu reagieren. Es benutzt fachliche Objekte und fachliche Logik, um Geschäftsprozesse abzubilden. Desweiteren benötigt ein PIS Dienste, die für die dauerhafte (persistente) Verwaltung der zu speichernden Daten in einem Datenbanksystem verantwortlich sind.

³⁹ Vgl. [DunkelHolitschke2003], 17

Beim Model-View-Controller werden diese Teilbereiche voneinander getrennt und einzelnen Softwareschichten zugeordnet. Die Model-Schicht enthält die darzustellenden Informationen, die Präsentationsschicht (View - alle für die Darstellung zuständigen Daten) und die Controller-Schicht (alle für die Steuerung der Software entwickelten Softwaremodule).

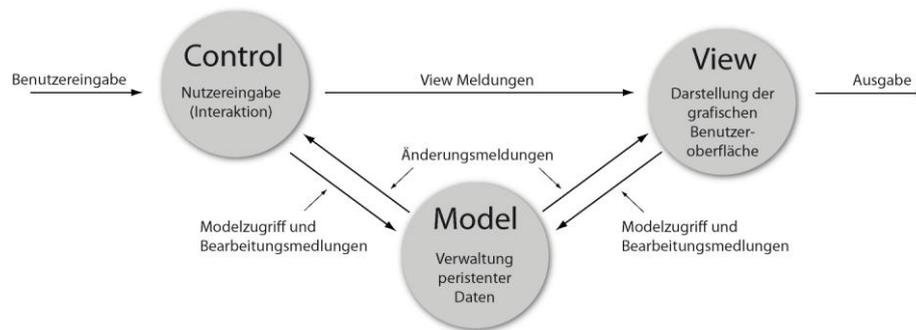


Abbildung 7: Model-View-Control (Schema)

Das MVC-Architekturmuster trifft keine Aussage über die Positionierung der Geschäftslogik innerhalb der MVC-Klassen. In welchen Teilbereichen der Anwendung Verarbeitungs- und Berechnungskomponenten eingebunden werden, ist unter anderem von der technischen Beschaffenheit abhängig. Werden beispielsweise in einer Client-Server-Architektur die Präsentations- und Steuerschicht auf den Arbeitsplatzrechner ausgelagert und die Anwendungsinformationen auf einem dezentralen Rechner gespeichert, sollten Berechnungsroutinen auch auf diesem ausgeführt werden. Nutzerendgeräten mit geringer Leistung kann so eine optimale Verarbeitung durch den leistungsfähigeren Server gewährleistet werden. In diesem Fall würde die Geschäftslogik größtenteils in die Model-Schicht ausgelagert.

2.2.3 Die Mikroarchitektur

Betrachtet man in der Makroarchitektur das allgemeine Zusammenspiel von Komponenten und Teilsystemen, so werden in der Mikroarchitektur tragfähige Strukturen für die einzelnen Softwaremodule entwickelt. Die einzusetzenden Programmiersprachen spielen dabei zunächst keine Rolle. Es werden detaillierte Grundstrukturen der einzelnen Softwaremodule festgelegt und so ein Ausgangspunkt für dessen Implementierung geboten. Die Entwicklung der Mikroarchitektur wird durch eine Vielzahl von sogenannten Designmustern (design pattern) unterstützt, welche ein spezielles und sehr eingegrenztes Entwurfsproblem behandeln. Eine detaillierte Erläuterung einzelner Designmuster ist in dieser Arbeit nicht vorgesehen.

Beschrieben wird die Mikroarchitektur in objektorientierten Systemen mittels der UML. Einzelne Teilbereiche werden durch Klassendiagramme und ihre Beziehungen, die dynamische Struktur durch Sequenz-, Aktivitäts- und Zustandsdiagramme dargestellt.

2.2.4 Die Technische Architektur

In der technischen Architektur wird die Mikroarchitektur den Bedingungen der eingesetzten Programmiersprachen angepasst. Die Mikroarchitektur wird um die entsprechenden Informationen ergänzt (Annotation), um sie mit entsprechendem Quellcode realisieren zu können. Die technische Architektur wird durch ein realisierungsnahes Klassendiagramm beschrieben und kann unmittelbar in Programmcode umgesetzt werden.

2.3 Anwendung des Phasenmodells auf den Entwicklungsprozess eines Personalinformationssystems in dieser Arbeit

Ziel dieser Arbeit ist, einen kompakten Einstieg zur Entwicklung eines PIS darzustellen. Da Anwendungen der Personalverwaltung in unzähligen Formen erstellt und angepasst werden können, werden lediglich die wichtigsten Entwicklungsschritte erfasst. Es wird bewusst die Zielgruppe von Entwicklern angesprochen, welche einen Einstieg in die Thematik Personalinformationssysteme suchen und allgemeine Konzeptions- und Entwicklungsansätze finden wollen. Zusätzliche betriebsbedingte Programmbereiche müssen durch den Entwickler eines Unternehmens eigenständig erkannt und implementiert werden.

Die folgenden Kapitel dieser Arbeit erfüllen den Zweck, die Grundlagen der Projektplanung, -konzeption und -entwicklung eines PIS in allgemeiner Form darzustellen. Hardwarebedingte Einschränkungen, koordinierende Aufgaben bezüglich mehrerer Projektgruppen und die zeitliche Abschätzung einzelner Projektphasen werden nicht berücksichtigt. Die besprochenen Themen beschränken sich nicht auf eine reine Aufzählung entwicklungsrelevanter Inhalte, sondern beinhalten bewusst Hinweise, allgemeine Beispiele und Wertungen, die dem Projektleiter bzw. dem Softwareentwickler bei der Erstellung eines auf den Betrieb zugeschnittenen Personalinformationssystem unterstützen können.

Die allgemeinen Phasen bei der Erstellung eines PIS werden neben Planung, Konzeption und Durchführung durch die qualitätssichernden Aufgaben bei Fertigstellung eines solchen Projektes beschrieben. Wie in Abbildung 7 dargestellt, werden in dieser Arbeit eine allgemei-

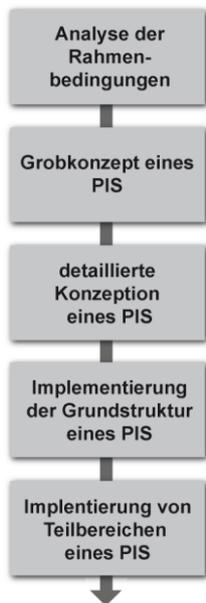


Abbildung 8: Phasen des Entwicklungsprozesses eines PIS in dieser Arbeit

ne Analyse der Rahmenbedingungen, das Grobkonzept eines PIS, ergänzt durch die detaillierte Konzeption einzelner Teilbereiche, und ausgewählte Implementierungsschritte der Anwendung beschrieben. Um den Rahmen dieser Arbeit nicht zu überschreiten wird das Thema Qualitätssicherung nicht erläutert.

Analyse der Rahmenbedingungen (Architekturbasis)

Zur Analyse der Rahmenbedingung gehören alle zur Vorbereitung eines Projekts notwendigen Maßnahmen. Neben dem Definieren der Projektziele aus Nutzer sowie aus Entwicklersicht werden in dieser Phase alle zur Ausführung eines PIS relevanten Hilfsmittel diskutiert. Es gilt zu Analysieren, welche hard- und softwaretechnischen Voraussetzungen geschaffen werden müssen, um ein PIS zu entwickeln und dieses nach den gewünschten Anforderungen ausführen zu können.

Konzeption (Makro- und Mikroarchitektur)

In der Phase der Konzeption werden anhand einer Beispielanwendung die wichtigsten Kernbereiche eines Personalinformationssystems beschrieben. Es werden verschiedene Entwurfstechniken für ausgewählte Beispiele dargestellt und praxisorientierte Anwendungskonzepte der einzelnen Teilbereiche analysiert. Phasenspezifische Aspekte wie zum Beispiel Systemspezifikationen, Validierungsprozesse, Lasten- und Pflichtenheft gehören zweifellos zu den wesentlichen Elementen der Konzeption eines PIS, liegen aber außerhalb der für diese Arbeit vorgesehen Grenzen.

Implementierung

Bei der Implementierung eines PIS werden alle programmiertechnischen Aufgaben durchgeführt und die fertigen Softwarekomponenten für den letztendlichen Endgebrauch vorbereitet. Angelehnt an die Gliederung der Konzeption werden im Kapitel der Realisierung die wichtigsten Softwaremodule erfasst, Standardprobleme in allgemeiner Form diskutiert und grundsätzliche Optimierungsmethoden verschiedener Entwicklungsprozesse analysiert.

3 Analyse der Rahmenbedingungen

Eigene Erfahrungen haben gezeigt, dass jedes Projekt einmalig ist. Jedes Projekt hat seine eigenen Spezifikationen und benötigt eine genaue Analyse der Projektziele. Die zu analysierende Faktoren und die dadurch erfassten Informationen bilden die Grundvoraussetzung für eine detaillierte Konzeption und die spätere Entwicklung der Applikation. Im folgenden Kapitel werden verschiedene Aspekte bezüglich der zu verarbeitenden Informationen sowie des Einsatzbereichs eines PIS dargestellt.

3.1 Die Hardwarekomponenten und die Softwarearchitektur

3.1.1 Die Hardwarekomponenten

Die Hardware beschreibt die Gesamtheit aller technischen Komponenten, die für die computergestützte Informationsarbeit erforderlich sind. Dem Einsatzbereich entsprechend können bei richtiger Analyse der notwendigen Hardwarevoraussetzungen Kosten gespart werden ohne ersichtliche Performance-Verluste des Personalinformationssystems zu erhalten.

Die Gesamtheit der technischen Komponente eines Personalinformationssystems umfasst die Rechner, die Peripherie und die Rechnernetze.

Rechner

Durch die typischen Einsatzbereiche von Rechnern kann in Zentral- und Arbeitsplatzrechner unterschieden werden. Aufgrund der schnellen Entwicklung wird auf die Angabe technischer Leistungsmerkmale (Prozessor, Arbeitsspeicher, Rechnerleistung, etc.)⁴⁰ verzichtet.

Zentralrechner (Server, Host): Der Zentralrechner wurde bis Ende der 80er Jahre⁴¹ als zentrales Mehrbenutzersystem in der betrieblichen Datenverarbeitung genutzt. Seit der Einführung von Rechnernetzen wurden derartige Systeme größtenteils abgeschafft und durch sogenannte Server als zentrale netzwerksteuernde Geräte ersetzt. Der technische Schwerpunkt liegt weiterhin bei der Berechnung und Verarbeitung von Daten. Es handelt sich um einen besonders leistungsfähigen Computer, welcher gleichzeitig vielen Benutzern (Klienten)

⁴⁰ Vgl. [Rohweder1996], 89 ff.

⁴¹ Vgl. [TraegerVolk2002], 257

zur Verfügung steht und als essentielles Instrument die gemeinsame Nutzung mehrerer Anwendungssysteme oder Programme realisiert.

Arbeitsplatzrechner (Client): Das Benutzen dieser Rechner erfolgt an einem festgelegten Arbeitsplatz. Meist als Personal Computer (PC) im Einsatz sollte der Client mit einer sehr komfortablen, grafikorientierten Benutzerschnittstelle ausgestattet sein. Oftmals befindet sich ein breites Spektrum an Standardsoftware auf diesen Rechnern, welche unabhängig vom Server als Standalone-Anwendungen genutzt werden. In Verbindung mit einem zentralen Rechner dienen diese Rechner jedoch weniger zur Berechnung verschiedener Anwendungsprozesse, sondern hauptsächlich der Übertragung und Darstellung von Daten.

Peripherie

Die Peripherie umfasst alle technischen Geräte der Eingabe- und Ausgabefunktionen. Die Eingabe erfolgt in der Regel unmittelbar über die Tastatur, die Maus und mit steigender Bedeutung auch über den Monitor. Kommen sogenannte Touchscreens zum Einsatz muss auf die Erstellung einer anwendbaren Arbeitsoberfläche geachtet werden. Ausgabefunktionen werden entweder direkt am Bildschirm oder über den Drucker ermöglicht. Der Monitor ist im Zuge der hohen Bedeutung grafikorientierter Benutzeroberflächen das wichtigste Ausgabe-medium. Der Einsatz von Druckern erfordert meist eigene Ausgabeformate und damit das Erstellen zusätzlicher speziell angepasster Ausgabefunktionen.

Rechnernetze

Rechnernetze gewährleisten durch ein Datenübertragungssystem die Verbindung zwischen mehreren Arbeitsplatz- bzw. Zentralrechnern. Befindet sich das Netzwerk innerhalb eines Gebäudes oder eines abgeschlossenen Firmengeländes, ist bei den meisten Unternehmen ein lokales Netzwerk (LAN) vorhanden. Die Verbindung der Rechner kann durch unterschiedliche Netzstrukturen (Netztypologien⁴²) erfolgen. Weitere Realisierungsmöglichkeiten einer Client-Server-Architektur ergeben sich durch das drahtlos Netzwerk (WLAN) oder dem Zugriff auf einen externen Webserver mittels Internetzugang. Übertragungsgeschwindigkeiten und zusätzliche Sicherheitsaspekte⁴³ zum Schutz der zu übertragenden Daten sollten, bei beiden Varianten unbedingt beachtet werden.

⁴² Vgl. [SchmidtHompelTen2010], 164

⁴³ Vgl. Kapitel 4.1.2 Sicherheit

Einsatzbereiche der Hardwarekomponenten

Werden im Bereich idiosynkratischer Prozesse⁴⁴ oder bei fehlender Netzwerkstruktur einfache Standalone-Anwendungen als Softwarelösung genutzt, ist der Einsatz eines Personalcomputers mit einer komfortablen, grafikorientierten Benutzerschnittstelle völlig ausreichend. Die Performance eines solchen Rechners ist nach heutigen Standards⁴⁵ nicht mehr ausführungsrelevant, sondern führt lediglich zu gering unterschiedlicher Dauer einzelner Lade- und Berechnungsprozesse. Für die Planung aufwendigerer innerbetrieblicher Arbeitsabläufe, ist der Einsatz von Standalone-Anwendungen jedoch veraltet.

Client-Server-Architekturen⁴⁶ lassen einen größeren und flexibleren Anwendungsbereich für den Gebrauch eines Personalinformationssystems zu. Bei dieser Variante speichern ein oder mehrere zentrale Server Datensätze und Anwendungsprogramme, welche von dem dezentralen Arbeitsplatzrechnern abgerufen und weiterverarbeitet werden können. Der Zugang kann über das firmeneigene Intranet oder browserbasierend über das Internet erfolgen. Wird mit einem Webserver gearbeitet, ist die Anwendung von jedem internetfähigem Rechner auch außerhalb der Firma erreichbar. Bei dieser Variante gilt es wichtige Sicherheitsaspekte⁴⁷ zu beachten, welche die Dauer der Entwicklungs- und Verarbeitungszeit der zu erstellenden Software erhöhen.

Für die Wahl der passenden Hardwarekomponenten gilt es ein weiteres wichtiges Kapitel der Softwareentwicklung zu betrachten – die Softwarearchitektur.

3.1.2 Softwarearchitektur

Anhand der Softwarearchitektur wird der grundlegende Aufbau eines Personalinformationssystems festgelegt und somit die Grundlage für die gesamte Systementwicklung geschaffen. Softwarearchitekturen können als fundamentaler Bauplan verstanden werden und gelten als entscheidender Erfolgsfaktor bei der Projektdurchführung. Eine gute Softwarearchitektur hilft

⁴⁴ Idiosynkratische Anwendungen beschreiben charakteristische Prozesse, die in ihrer Art nur mit einem geringen und an spezielle Situationen angepassten Aufwand verbunden sind. Wesentliche Beispiel sind: „*Organisation von Sportveranstaltungen, Sitzplangestaltung bei Veranstaltungen, Zuordnung von Parkflächen, etc.*“.

⁴⁵ Vgl. PC-Komplettsysteme 2004: „*Dell Dimension 4700 D1091, Acer Veriton 7700G, HP Business Desktop DX6050, IBM Thinkcentre S50, Sony Vaio PCVRS602*“. Detaillierte Angaben sind beim jew. Hersteller erhältlich.

⁴⁶ Vgl. [Lassmann2006], 147 ff.

⁴⁷ Vgl. Kapitel 3.2.1 Sicherheit

die Komplexität großer Softwaresysteme zu beherrschen, die Qualität der Software zu verbessern und eine übersichtliche Organisation der Projektdurchführung zu erstellen.

Verteilte Systeme

Bei Personalinformationssystemen gibt es im Normalfall mehrere Benutzer, die gleichzeitig auf denselben Datenbestand zugreifen, Informationen austauschen oder Systemdienste nutzen wollen. Zusätzlich besitzen die in einer Anwendung verwendeten Softwarekomponenten unterschiedliche Leistungsansprüche und sollten möglichst auf für sie geeignete Endgeräte verlagert werden. Ein derzeit aktuelles Beispiel ist der verstärkte Gebrauch von Software auf Nutzerendgeräten mit einem sehr geringen Leistungsvermögen, wie beispielsweise Mobiltelefone und PDAs (personal digital assistants). Diese sind darauf angewiesen, dass rechenintensive Dienste auf anderen leistungstärkeren Rechnern angeboten werden.

Moderne Systeme sind aus diesem Grund meist auf mehreren Rechnern verteilt, welche über ein Netzwerk miteinander kommunizieren. Ein verteiltes System lässt sich gut an die Leistungserfordernisse anpassen und ermöglicht ein dezentrales Abrufen einzelner teilweise von einander unabhängiger Softwarekomponenten. Fällt eine Teilkomponente aus, sollten alle anderen Dienste aufrecht erhalten und so das Gesamtsystem verfügbar bleiben.

Damit Software auf mehreren Rechnern verteilt werden kann, muss sie eine geeignete Struktur besitzen. Eine gute Basis bietet die in Kapitel 2.2.2 beschriebene Schichtenarchitektur. Je nach Anforderungen bzw. eingesetzten Technologien lassen sich die einzelnen Schichten auf mehrere Rechner verteilen. Dabei kann zwischen schwer- und leichtgewichtigen Klienten unterschieden werden.

Standardarchitekturen verteilter Systeme

Schwergewichtige Klienten (Fat Clients) - Bei schwergewichtigen Klienten befinden sich alle Schichten auf dem Arbeitsplatzrechner. Nur die eigentliche Datenhaltung in einem Datenbanksystem wird auf zentralen Servern realisiert. Der Zugriff auf die Datenbank erfolgt über durch den Arbeitsplatzrechner veranlasste Befehle, welche über ein entsprechendes Netzwerkprotokoll gesteuert werden.

Abbildung 8 zeigt eine Architektur mit schwergewichtigen Klienten. Alle Softwareschichten befinden sich auf dem Arbeitsplatzrechner. Diese umfassen die GUI-Komponenten der Präsentationsschicht, die fachlichen Objekte der Anwendungsschicht und die Persistenzschicht,

um gespeicherte Daten abzurufen. Lediglich das für die eigentliche Datenhaltung genutzte Datenbanksystem befindet sich auf einem eigenen Server.

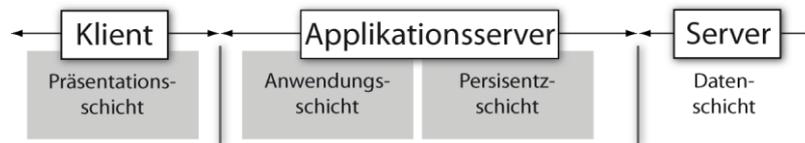


Abbildung 9: Fat-Client-Architektur

Derartige Architektursysteme sind relativ leicht zu implementieren, da lediglich der Datenbankzugriff über das Netzwerk erfolgt. Alle weiteren Funktionalitäten werden lokal ausgeführt und sind deshalb technisch einfacher zu realisieren.

Nachteilig wirkt sich dagegen die hohe Abhängigkeit des Systems von den zur Verfügung stehenden Arbeitsplatzrechnern aus. Da alle Applikationen bis auf die Datenbankabfragen auf dem Client ausgeführt werden, müssen die genutzten Endgeräte entsprechend leistungsstark sein, wodurch unter Umständen kleine Geräte ohne ausreichende Rechenleistung nicht eingesetzt werden können. Zudem sind Fat-Client-Systeme schlecht skalierbar, das heißt sobald Performanceprobleme auftreten, müssen alle betreffenden Arbeitsplatzrechner leistungstechnisch aufgerüstet werden. In Unternehmen mit einer hohen Zahl an Klienten kann dies zu enormen Kosten führen. Ebenso wie bei der Softwareverteilung. Bei einer neuen Version des PIS muss für jeden Arbeitsplatzrechner in einem sehr zeitaufwändigen Prozess das aktuelle Programm installiert werden.

Leichtgewichtige Klienten (Thin Clients) - Bei leichtgewichtigen Klienten wird nur die Präsentationsschicht auf dem Arbeitsplatzrechner ausgeführt. Die Anwendungsschicht, das heißt die fachlichen Objekte und die Geschäftslogik, befindet sich auf einem oder mehreren eigenen Rechnern den sogenannten Applikationsservern. Bei einem Ereignis in der Präsentationsschicht wird eine Funktion der Anwendungsschicht aufgerufen und dieser Daten aus der Benutzeroberfläche übergeben. In der Anwendungsschicht wird dadurch ein Geschäftsprozess ausgelöst, der Daten sucht oder ändert und diese anschließend an die Präsentationsschicht zurück gibt. In dieser werden die erhaltenen Informationen implementiert und angezeigt.

Abbildung 9 zeigt eine Thin-Client-Architektur. Es werden dieselben logischen Schichten mit identischen Verantwortlichkeiten, wie bei Fat-Client-Systemen genutzt, diese aber in anderer Weise physikalisch verteilt. Auf dem Arbeitsplatzrechner befinden sich nur die Komponenten der Präsentationsschicht, die durch definierte Schnittstellen auf die Anwendungsschicht zu-

greifen. Deren Softwaremodule werden auf einem Applikationsserver hinterlegt, wodurch Verarbeitungsroutinen und Datenbankabfragen serverseitig ausgeführt werden.

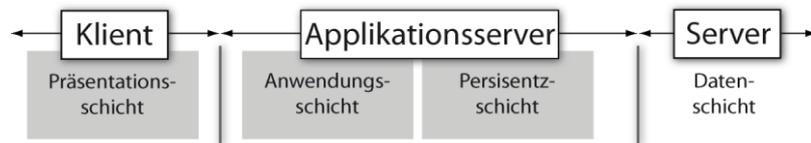


Abbildung 10: Thin-Client-Architektur

Das Ergänzen rechenintensiver fachlicher Prozesse verlangt bei Thin-Client-Systemen lediglich das Aufrüsten der Applikationsserver. Außerdem bedeuten Änderungen in der Anwendungsschicht keine Neuinstallationen auf dem Arbeitsplatzrechnern, da diese nur die Präsentationsschicht enthalten, wodurch auch leistungsschwache Geräte eingesetzt werden können.

Die Implementierung derartiger Architekturen ist aufwendig, weil komplexere Technologien genutzt und zusätzliche Schnittstellen definiert werden müssen. Desweiteren ist zu beachten, dass sehr viele Zugriffe über das Netz erfolgen und damit verbundenen Ladeprozesse langsamer als bei der lokalen Verarbeitung sind.

Internetbasierte Systeme - Eine besondere Rolle spielen internetbasierte Systeme. Die Architektur besteht ausschließlich aus leichtgewichtige Klienten, mit einer durch einen Browser angezeigten grafischen Benutzeroberfläche. Die komplette Software befindet sich auf dem Server, wodurch die einzelnen Arbeitsplatzrechner auch als Ultra Thin Clients bezeichnet werden.

Die Benutzeroberfläche muss auf dem Server dynamisch erzeugt, in ihr die betreffenden geschäftsprozessbezogene Daten aufgenommen und als Webseite auf dem Arbeitsplatzrechner angezeigt werden. Diese enthalten sowohl statische Anteile, wie Bilder, Schaltflächen und Texte, als auch dynamisch einzufügende Daten, wie beispielsweise die Ergebnisse einer Suchanfrage. Abbildung 10 zeigt das Strukturbild einer solchen Architektur. Im Gegensatz zur Thin-Client-Architektur erhält der Applikationsserver bei Web-Server-Architekturen eine weitere Softwareschicht, die „Serverpräsentationsschicht“, die für die dynamische Erzeugung der Webseiten und die durch Benutzerereignisse ausgelöste Steuerungsprozesse verantwortlich ist. Die Webseiten sind in einer Markup-Sprache beschrieben, welche in dieser Arbeit zum Einbinden von SWF-Dateien, dem Dateiformat zur Darstellung von Adobe Actionscript Applikationen, genutzt wird.



Abbildung 11: Web-Server-Architektur

Web-Server-Architekturen haben den großen Vorteil komplett Arbeitsplatz unabhängig zu sein. Sie sind jedoch immer von der Datendurchsatzrate des Netzwerkes abhängig. Bei einer langsamen Internetverbindung und einer großer Anzahl an Ladevorgänge kann diese Systemarchitektur schnell an seine technischen Grenzen gelangen.

Das Architekturdesign eines PIS

Zur Umsetzung des Personalinformationssystems wird ein dezentrales System empfohlen. Die Objekte der Präsentationsschicht sollten als selbstständige Anwendung, entweder als ausführbare Datei oder ein in den Browser integriertes Modul, auf die Arbeitsplatzrechner geladen und dort ausgeführt werden.

Für die Implementierung der clientseitigen Applikationen wird im Rahmen einer Beispielanwendung das von Adobe entwickelte SWF-Format genutzt. SWF bietet neben einem sehr guten Sicherheitskonzept die Unterstützung von Flash/Actionscript 3.0, was sich wiederum durch zahlreiche quillcodebasierende grafische Funktionen auszeichnet.

Applikationen der Anwender- und Persistenzschicht befinden sich auf einem zentralen PHP5 und MySQL 5.1 fähigem Apache-Server. Neben einer MySQL-Datenbank zur Speicherung der Datenbasis sind hier alle fachlichen Objekte zur Umsetzung serverseitiger Verarbeitungsroutinen des Personalinformationssystems untergebracht.

Die Zusammenarbeit von Klient und Applikationsserver erfolgt durch eine von Actionscript 3.0 realisierte Schnittstelle, welche im Laufe dieser Arbeit⁴⁸ detailliert erklärt wird.

3.2 Besondere Programmanforderungen durch den Benutzer

Der Auftraggeber, und damit in den meisten Fällen der Endverbraucher, ist der wichtigste Faktor beim Konzipieren einer Software. Es gilt genau zu erfassen, welche Projektziele die-

⁴⁸ Vgl. Kapitel 5.2.1 Die Schnittstelle Actionscript/PHP (URLRequest, URLLoader, URL-Variables)

ser verfolgt und welche erweiterten Programmmodule notwendig sind. Zu diesem Zweck existieren einzelne Analyseschwerpunkte die im Folgenden beschrieben werden.

3.2.1 Sicherheit

Der Persönlichkeitsschutz von Mitarbeitern stellt wichtige Anforderungen an Personalinformationssysteme. Nicht nur die gesetzlichen Vorgaben sondern auch das notwendige Vertrauen des Benutzers in die Software verlangen klare Richtlinien für den Umgang mit Mitarbeiterdaten. Nur der Mitarbeiter selbst und durch den Betriebsrat berechnigte Personen sollten auf personengebundene Daten zugreifen können. Die Art der Anwendung und der Einsatzbereich fordern in den meisten Fällen speziell an das Unternehmen und der dort vorherrschende Infrastruktur angepasste Sicherheitsvorkehrungen.

Eine an dieser Stelle häufig genutzte Variante zum Schutz vor Fremdzugriffen ist ein beim Start der Software genutzter Anmeldebereich. Einmalig melden sich die Anwender mittels Benutzername und Passwort oder einem eindeutig identifizierbaren Zugangsschlüssel für den Gebrauch der Software an. Durch das anschließende Erkennen der zugehörigen Zugriffsrechte können personenorientierte Daten gezielt zur Verfügung gestellt werden.

Bei der Übertragung von Informationen entstehen weitere Sicherheitslücken, welche unbefugten Personen die Möglichkeit geben, Daten abzufangen und für eigene Dienste weiterzuverarbeiten. Verschiedene technische Lösungen zur Vermeidung derartiger Fremdzugriffe werden im Kapitel 3.3.3, 5.2.1 und 5.2.4 detailliert beschrieben. Um inadäquate Entwicklungsschritte zu vermeiden, gilt es im Rahmen der Konzeption abzuschätzen, welcher Sicherheitsgrad angebracht ist.

3.2.2 CMS - Weiterentwicklung

Der ständige Wandel von betrieblichen Gegebenheiten kann zu immer neuen Ansprüchen an das Personalinformationssystem führen. Diese in der Regel isolierte Aufgabe der Softwareweiterentwicklung durch den Programmierer oder dem Webmaster führt zu einer dauerhaften Kostenstelle und einer entwicklungstechnischen Abhängigkeit. Mittels Content Management System kann ein Großteil inhaltlicher Programmänderungen auf verschiedene Organisationseinheiten eines Unternehmens verteilt werden. Diese erstellen, pflegen, verteilen und

löschen Content-Objekte im laufenden Betrieb und benötigen zur effizienten Abwicklung und Verwaltung ihrer Aufgaben verbindliche Prozesse und darauf abgestimmte Softwaremodule.

Definition Content Management System

Im strengen Sinn der Definition werden innerhalb eines Content Management Systems (CMS) sämtliche digitale Daten eines Softwareprodukts, auf die die Teilnehmer eines Intranets oder dem Internet gemeinsam zugreifen können, organisiert und verwaltet. Neben diesen Aufgaben beinhalten Content Management Systeme zusätzliche Funktionen zur Erstellung, Präsentation und Kontrolle dieser Daten.

Systemfunktionen eines CMS

Das Prinzip eines Content Management Systems umfasst die Trennung von Softwarekomponenten bezüglich der Struktur, dem Inhalt und der Darstellung⁴⁹. Die Struktur definiert die Anordnung, die Abfolge und die Verschachtelung der zu verarbeitenden Einzelinformationen. Mit der Darstellung wird eine formale Beschreibung zur Abbildung von Rohinhalten auf einem möglichen Ausgabemedium beschrieben. Oft werden Stylesheets, welche informationstechnische Anweisungen über Formatierung und Positionierung des Inhalts enthalten, als Entwicklerwerkzeug genutzt. Der Inhalt beinhaltet die für den Anwender der Software relevanten Daten. Durch diese Dreiteilung verfügen CMS gestützte Programme über vielfältige Möglichkeiten zur weiteren, clientseitigen und automatisierten Datenverarbeitung. Neben der Darstellung von Informationen auf unterschiedlichen Ausgabemedien können Daten unter anderem vom Entwickler unabhängig Benutzer- und bedarfsgerecht aufbereitet werden.

Folgende **Systemfunktionen** beschreiben typische Merkmale eines CMS:

- Getrennte Bearbeitung von Struktur, Darstellung und Inhalt
- Dynamisches Einbinden von Rohinhalten in die Darstellungsvorlagen (Stylesheets) für unterschiedliche Ausgabemedien bzw. Benutzerprofile
- Unterstützende Aufgaben bei der Formatierung von Rohinhalten durch standardisierte und webbasierte Templates
- Entwicklerunabhängiges Verwalten von Struktur- und Darstellungsinformationen
- Automatisierung der Pflege (Löschen, Verschieben und Ändern von Inhalten)
- Eigenständige Sicherungs- und Archivierungsfunktionen

⁴⁹ Vgl. [BullSchustWilh2000], 8

- Anbindung und Beteiligung dezentral agierender Contentlieferanten

Durch die Tatsache, dezentral arbeitende Mitarbeiter in die Weiterentwicklung der Software einbinden zu können, wird auf Grund der geringeren Durchlaufzeiten eine höhere Aktualität programmrelevanter Inhalte gewährleistet. Desweiteren werden Formatierungsfehler und fehlerhafte Übertragungen beim Anwenden von sogenannten Medienbrüchen⁵⁰ reduziert.

Das Generieren CMS basierender Funktionen bedeutet einen zusätzlichen Entwicklungsaufwand, durch welchen die Erstellungskosten einer Software stark steigen können. Damit ein CMS gestütztes System ordnungsgemäß eingesetzt werden kann, sind zusätzliche Softwarekomponenten erforderlich. Beim Planen und Konzipieren der Anwendung sollte unbedingt abgeschätzt werden, ob sich der zusätzlichen Entwicklungsaufwand mit den damit verbundenen Mehrkosten gegenüber der bei Gebrauch kostensparenden entwicklerunabhängigen Dynamik lohnt.

3.2.3 Benutzergruppen

Benutzerprofile und die damit verbundene Rechteverwaltung dienen der Zugriffsordnung der Anwender auf verschiedene Programmsegmente und -inhalte. Gerade durch den Schutz der Persönlichkeitsrechte muss eindeutig geregelt sein, welche Mitarbeiterinformationen, CMS-Funktionen und Steuerungsprozesse für den Benutzer sichtbar und veränderbar sind. Durch das Erstellen und Speichern von Benutzerprofilen kann dieses Problem gelöst werden. In der Regel kommen die fünf Benutzergruppen Mitarbeiter, Teamleiter, Operator, Admin und benutzerdefinierte Profile zum Einsatz.

Der **Mitarbeiter** bekommt die Möglichkeit seine persönlichen Daten zu sehen, sich selbst in laufende Steuerungsprozesse einzuordnen und zukünftige Termine, wie beispielsweise Urlaubstage oder Weiterbildungen, beantragen zu können. Die Einsicht auf mitarbeiterfremde Informationen oder Änderungen persönlicher Daten sind ihm in den meisten Fällen nicht gestattet.

Dem **Teamleiter** ist in der Regel ein kleiner Mitarbeiterkreis zugeordnet. Er bekommt die persönlichen Daten ihm zugeteilter Personen zu sehen und kann diese überarbeiten. Die

⁵⁰ Die Übertragung von Informationen eines Mediums in ein Anderes wird als Medienbruch bezeichnet. Oft entstehen bei einem Medienwechsel Darstellungsfehler, welche durch CMS-Funktionen reduziert werden.

Vgl. [FeldbrBrechtHadra2008], 45

laufende Einteilung der Steuerungsprozesse kann in einem gewissen Rahmen durch den Teamleiter bestimmt oder verändert werden. Von Mitarbeitern beantragte Termine werden durch den Teamleiter bestätigt bzw. abgelehnt. Desweiteren kann dieser neue Anwender mit Mitarbeiterstatus für seine Gruppe anlegen und wieder entfernen.

Der **Operator** ist dafür zuständig, die für Steuerungsprozesse notwendigen Hintergrundinformationen anzulegen. Die Steuerungsdaten werden von ihm angelegt, verändert und eingesetzt. Er hat außer auf die eigenen Daten keine Einsicht auf Informationen anderer Personen. Um seine Arbeit zu erleichtern, kann das PIS durch das Speichern und Laden verschiedener Standardszenarien unterstützend eingesetzt werden.

Der **Admin** hat vollen Zugriff auf alle Programmsegmente, welche in Bezug auf den Mitarbeiter, den Teamleiter und den Operator durch weitere Funktionen ergänzt werden. Im Rahmen der Rechteverwaltung kann er den Status des Mitarbeiters beliebig setzen oder verändern. Programmspezifische Änderungen werden von ihm vorgenommen. Es muss mindestens ein Anwender mit Admin-Status vorhanden sein.

Der Einsatz **benutzerdefinierter Profile** ist immer dann notwendig, wenn sich kein für den Anwender zuweisbares Profil verwenden lässt. Es muss im Rahmen der Entwicklung genau analysiert werden, welche Softwaresegmente existieren und auf den Anwender zu projizieren sind. Durch den Admin bzw. den Teamleiter kann durch diesen Prozess ein genau auf den Nutzer zugeschnittenes Benutzerprofil erstellt werden.

Um die Entwicklungskosten eines PIS möglichst gering zu halten, sollte genau analysiert werden, ob, welche und wie viel verschiedene Benutzerrechtstypen zum Einsatz kommen.

3.3 Verwendete Technologien

Im Rahmen dieser Arbeit werden die Technologien MySQL zur Speicherung von Daten, PHP für Applikationen der Anwender- und Persistenzschicht und Flash/Actionscript für die Umsetzung der grafischen Benutzeroberfläche verwendet. Was sich hinter diesen Begriffen verbirgt, warum die Wahl gerade auf diese drei Technologien gefallen ist und welche allgemeinen Optimierungsmethoden möglich sind, wird in den folgenden Kapiteln erklärt.

3.3.1 MySQL zur Speicherung von Daten

Die Datenbank bildet das Fundament eines jeden Personalinformationssystems. In ihr werden Informationen gespeichert und stehen zum Abruf bzw. zur Verarbeitung bereit. Es existiert mittlerweile eine Vielzahl von Datenbankmanagementsystemen⁵¹, die in ihrer Art für viele spezielle Anwendungsbereiche eingesetzt werden können. Die Hauptmerkmale unterscheiden sich dabei in Funktionsumfang, Performance, Stabilität und Kosten.

Das Datenbanksystem MySQL

MySQL gilt als das am weitesten verbreitete⁵² frei erhältliche Datenbankmanagementsystem und ist nicht nur schnell, stabil und leicht zu erlernen sondern unterstützt auch alle gängigen Betriebssysteme (Windows, Linux, Mac OS X, diverse UNIX-Derivate). Es kann in allen erdenklichen Programmiersprachen (z.B. C, C++, C#, Java, Perl, PHP, VB und VB.net) erstellt werden und ist für viele Anwendungen kostenlos verfügbar. Tabellen können mit MySQL angelegt und mit der kostenlosen Datenbankkonsole XAMPP⁵³ bearbeitet werden. Die nachfolgende Übersicht projiziert die in der Beispielanwendung zu erfüllenden Kriterien auf MySQL.

Steuerung per PHP	Es gibt Zugriffsbibliotheken für C, C++, Java, Perl, PHP, Python und TCL
Speicheroptimierung durch Datentypdefinition	Sehr viele Datentypen stehen zur Verfügung: Integer mit oder ohne Vorzeichen und 1, 2, 3, 4 oder 8 Bytes Länge, Float, Double, Text, BLOB ⁵⁴ , Date, Time, Set und Enum Typen, nur um einige von ihnen aufzuzählen.
ausreichende Tabellengrößen	Bis zu 16 Indizes pro Tabelle sind möglich. Jeder Index darf aus 1 bis 15 Spalten bestehen. Die maximale Indexlänge beträgt 256 Bytes.
Betriebssystem Windows 7	MySQL unterstützt viele Betriebssysteme, unter anderem Linux, Solaris, Windows.
Erweiterungsmöglichkeiten der Datensätze	Datensätze fester und variabler Länge werden unterstützt.
optimale Abfragegeschwindigkeit	Alle SQL-Funktionen sind geschwindigkeitsoptimiert.

Abbildung 12: Zu erfüllende Kriterien von MySQL in Bezug auf die Beispielanwendung

MySQL verkörpert ein sogenanntes relationales Datenbankmanagementsystem (RDBMS - relational database management system). RDBMS Datenbanken ordnen Daten in Tabellen

⁵¹ beispielsweise MySQL, Oracle, IBM DB2/UDB, PostgreSQL, Microsoft SQL Server uvm.

⁵² Vgl. [Kofler2005], 5

⁵³ Vgl. Kapitel 3.3.2 Apache, XAMPP und phpMyAdmin

⁵⁴ BLOB – Ein dem bekanntem String-Typ ähnlicher Datentyp mit maximal 65535 Zeichen (16 bit). Zusätzlich wird bei diesem Datentyp auf Groß- und Kleinschreibung geachtet.

an und verwalten zwischen den Tabellen bestehende Beziehungen (engl. relations). Das Personalinformationssystem ist auf diese Verbindungen angewiesen. Im Bereich der Steuerprozesse werden beispielsweise Runinformationen, Personendaten und Kalenderdaten ausgewertet und einzelnen Mitarbeitertabellen zugeordnet. Gleichzeitig schließen RDBM Systeme Performance mindernde Faktoren aus. Es werden nur Verknüpfungen angelegt, welche von der Anwendung aktiv genutzt werden. Auf Grund dessen werden Abfragen von RDBM Systemen schneller und effektiver als bei anderen Datenbankmodellen ausgeführt.

Andere heute üblichen Datenbanksysteme erfüllen die für das PIS notwendigen Kriterien nicht. So basiert das hierarchische Datenbankmodell auf einer Baumstruktur, was nur Abfragen zwischen Eltern- und Kindknoten ermöglicht. Abfragen zwischen einzelnen Tabellen sind nicht möglich. Ein Netzwerkdatenbanksystem erfüllt ähnlich zu RDBM Systemen viele für ein PIS notwendige Anforderungen, legt aber alle unter den Tabellen möglichen Verknüpfungen an. Abfragen werden durch das Nutzen nichtrelevanter Verknüpfungen verlangsamt, wodurch das Netzwerkdatenbanksystem eine suboptimale Lösung für ein PIS darstellt.

Mit der Wahl des Datenbanksystems, gilt es dieses unter performanceoptimierenden Gesichtspunkten einzurichten. Die folgenden Abschnitte zeigen einige grundsätzlichen Entwicklungskriterien und deren Anwendung auf das Personalinformationssystem.

Speicher-Engines (Tabellentypen)

MySQL unterstützt verschiedene Tabellentypen⁵⁵ und Datensatzformate. Für jede Tabelle kann neu entschieden werden, welche Speicher- und Indizierungsmethode verwendet werden soll. Für das PIS sind ausschließlich zwei Speicher-Engines relevant.

Das MyISAM (My Indexed Sequential Access Method) Tabellenformat gewährleistet schnelle Zugriffe mit kleinen Datensätzen. Sicherheitsmaßnahmen zum Schutz der Daten werden jedoch vernachlässigt. Alle stattgefundenen Änderungen gelten als gesetzt, wodurch vorherige gespeicherte Informationen nicht durch die Anwendung von Sicherheitskopien wiederherstellbar sind.

Die InnoDB Speicher-Engine kann für große Datenmengen mit einem hohen Sicherheitsanspruch genutzt werden. Für Personalinformationssysteme sollten hauptsächlich Tabellen des InnoDB-Typs eingesetzt werden, da eine große Anzahl an Informationen verwaltet wird und die Sicherheit der gespeicherten Daten im Vordergrund steht.

⁵⁵ Vgl. [devmysqlcomSE2010]

Normierung relationaler Datenbanken

Große nichtnormalisierte Tabellen kosten Systemleistung. Um die Tabelle in eine für relationale Datenbanken passende Form zu bringen, werden Daten normalisiert. Mit den fünf Regeln der Normalisierung⁵⁶ können Datenbanken grundsätzlich optimiert werden.

Eliminierung sich wiederholender Gruppen: Für jede Gruppe sich wiederholender Daten wird eine eigene Tabelle angelegt. In diesen Tabellen wird ein eindeutiger Schlüssel erstellt, d.h. eine Spalte, deren Werte jeweils einmalig vorhanden sind.

Eliminierung redundanter Daten: Wenn eine Eigenschaft mehrere Werte annehmen kann, werden diese in einer eigenen Tabelle untergebracht.

Eliminieren von Spalten, die von keinem Schlüssel abhängen: Wenn Eigenschaften keinen Zusammenhang mit dem Schlüsselfeld haben, also nicht ebenso eindeutig zugeordenbar sind, werden diese in eine eigene Tabelle übertragen.

Isolieren von unabhängigen Beziehungen: Keine Liste darf zwei oder mehr Beziehungen haben, die nicht direkt abhängig sind.

Beziehungen definieren: Zwischen den so geschaffenen Tabellen definiert man Beziehungen. Es können sogenannte 1-n, 1-1 oder n-n Beziehungen je nach Verarbeitung des Referenzschlüssels genutzt werden.

Große Tabellen aufteilen

In machen Situationen ist es vorteilhaft häufig genutzte Tabellen in mehrere kleine Tabellen aufzuteilen. Dies vermindert zwar die Übersichtlichkeit der Datenbank, kann aber dafür sorgen, dass aus einer dynamischen Tabelle kleinere statische Tabellensegmente herausgelöst werden. Diese können dann isoliert für schnellere Abfragen genutzt werden.

Verwenden des jeweils kleinstmöglichen Datentyps

Zur Deklaration von Spalten bietet MySQL eine Vielzahl verschiedener Datentypen. Durch das exakte Anordnen des jeweils für die Anwendung notwendigen Spezialtyps, lassen sich

⁵⁶ Vgl. [Krause2005], 554 f.

Festplatten- und Arbeitsspeicher einsparen. Eine Spalte des kleineren MEDIUMINT-Typ benötigt beispielweise 25 Prozent⁵⁷ weniger Speicherplatz als der oft verwendete INT-Typ.

Das Nutzen von Indizes

Prinzipiell dienen Indizes dem schnellen Finden von Datensätzen mit vordefinierten Spaltenwerten. Ist ein Index deklariert, startet die Suchabfrage an der entsprechenden Stelle und nicht am Anfang der Tabelle. Indizes sollten nur angelegt werden, wenn diese auch wirklich Verwendung finden. Aufgrund der zusätzlichen Informationen verlangsamen Indizes den Schreibprozess, wodurch diese Methode nur beim Lesen einer Tabelle Vorteile bringt.

3.3.2 PHP5 für die Anwendungs- und Persistenzschicht

PHP ist eine Skriptsprache, die größtenteils zur Erstellung dynamischer Webseiten oder Webanwendungen genutzt wird. Die aktuelle Version PHP5 basiert auf einer an C angelehnten Syntax, zeichnet sich durch zahlreiche Funktionsbibliotheken aus und wird als lizenzfreie Software angeboten. Der große Funktionsumfang von PHP zeigt sich besonders bei den Datenbankfunktionen in Bezug auf die weitreichende Unterstützung von MySQL. Mit PHP5 wurde die bisher höchste Entwicklungsstufe veröffentlicht. Objektorientierte Anwendungen sind mit dieser Version im Vergleich zu anderen Programmiersprachen und den eigenen Vorgängerversionen einfacher und effizienter zu erstellen. Zusätzlich wird eine große Zahl an Erweiterungen für die Zusammenarbeit mit anderen Diensten, wie das Nutzen von Frameworks oder dem Netzwerkprotokoll SOAP⁵⁸, zur Verfügung gestellt. Schließlich weist PHP eines der wichtigsten Kriterien für eine Skriptsprache auf, es ist leicht zu erweitern. Wenn die Programmiersprache eine benötigte Funktion nicht bietet, kann die Unterstützung dafür selbstständig ergänzt werden. PHP-Code wird serverseitig angewendet. Der Quellcode wird durch einen Interpreter auf einem Webserver verarbeitet und erst dann an den Webbrowser oder die clientseitige Anwendung übermittelt.

Apache, XAMPP und phpMyAdmin

Zum Ausführen von PHP-Skripten muss ein entsprechend eingerichteter Server vorhanden

⁵⁷ Vgl. [devmysqlcomDS2010]

⁵⁸ SOAP (*Simple Object Access Protocol*) ist ein Netzwerkprotokoll zum Austauschen von Daten zwischen verschiedenen Systemen. SOAP stützt sich auf die Standards XML (Repräsentation) und dem TCP/IP-Referenzmodell (Übertragung).

sein. Der Apache HTTP Server, ein Produkt der Apache Software Foundation, gilt als der meistbenutzte⁵⁹ Webserver netzwerkbasierender Anwendungen. Er wird als lizenzfreie Software geführt und unterstützt alle derzeit verfügbaren Betriebssysteme. Der Apache-Webserver ist modular aufgebaut. Die einzelnen Module realisieren beispielsweise die Kommunikation zwischen Browser und Webserver, das Verschlüsseln von Daten und das Verarbeiten serverseitiger Skriptsprachen. Häufig verwendete Skriptsprachen sind PHP und Perl. Diese sind kein Bestandteil des Webservers, sondern müssen als eigene Module eingerichtet werden.

Mit XAMPP wird eine einfach zu bedienende Software zur Installation aller für eine Webanwendung notwendigen Servermodule angeboten. Die Kurzform XAMPP steht für den kombinierten Einsatz von Programmen auf betriebssystemunabhängiger Basis, um dynamische Webseiten zur Verfügung zu stellen. Dabei stehen die einzelnen Buchstaben für die verwendeten Komponenten: X – frei wählbares Betriebssystem, Apache, MySQL und PHP bzw. Perl. XAMPP wird in dieser Arbeit als grundlegendes Entwicklungstool genutzt, ist jedoch aufgrund unsicherer Voreinstellungen nicht als Server-Basis eines Personalinformationssystems zu empfehlen. Es wird für die Entwicklung der Beispielanwendung als eine Experimentalmgebung verwendet. Für den praxisrelevanten Einsatz wird zu sogenannten, in dieser Arbeit nicht näher beschriebenen, LAMP (unter Linux) bzw. WAMP (unter Windows) Systemen geraten, welche durch die spezielle Anpassung an das jeweilige Betriebssystem, wichtige Sicherheitseinstellungen unterstützen.

phpMyAdmin ist das wohl beliebteste Administrationswerkzeug für MySQL und als zusätzliches Php-Skript in XAMPP integriert. Datenbanken und Tabellen können skriptsprachenunabhängig erzeugt, verändert und wieder gelöscht werden. Daten bzw. ganze Datensätze können importiert und exportiert sowie diverse andere Administrationsaufgaben durchgeführt werden. Zusätzlich sind die angelegten Datenbanken in grafisch übersichtlicher Form zur Unterstützung konzeptioneller Aufgaben darstellbar. Die in den folgenden Kapiteln skizzierten Grafiken, welche die für ein Personalinformationssystem einsetzbaren Datensätze repräsentieren, sind beispielsweise ausschließlich mit Hilfe von phpMyAdmin erzeugt.

3.3.3 Flash CS4 für die grafische Benutzeroberfläche

Für die Erstellung der grafischen Benutzeroberfläche wird in dieser Arbeit die Entwicklungs-

⁵⁹ [Wolfgarten2004], 19

umgebung Flash CS4 genutzt. Flash CS4 ist ein Produkt des US-amerikanischen Softwareunternehmens Adobe Systems und ist im Vergleich zu PHP weder lizenzfrei noch eigenständig erweiterbar. Flash wird clientseitig angewendet und kann nur in Verbindung mit einem entsprechenden Browser Plugin oder einem auf dem Arbeitsplatz installierten Flash Player ausgeführt werden. Flash erfüllt somit keine der noch in den vorangegangenen Kapiteln als positiv dargestellten Eigenschaften, die zur Wahl der entsprechenden Software geführt haben. Das folgende Kapitel wird zeigen, warum diese Software trotzdem für die Erstellung eines Personalinformationssystems geeignet ist und welche Merkmale diese Entwicklungsumgebung auszeichnen.

Flash ist eine Entwicklungsumgebung zur Umsetzung multimedialer Inhalte. Ursprünglich wurde Flash für die Erstellung von Animationen genutzt. Zu diesem Zweck beinhaltet es eine filmähnliche Animationszeitleiste, in der die Steuerung von Bewegungen bildabhängig erfolgt. Zusätzlich bietet Flash ein eigenes Dateiformat namens „SWF“, welches ein speziell auf serverseitige Anwendung ausgerichtetes Sicherheitskonzept verfolgt. Ergänzend verfügt Flash über eine eigene Programmiersprache: Actionscript. Diese Sprache basiert auf dem ECMAScript-Standard (ECMA-262) und ermöglicht das Entwickeln komplexer, interaktiver Webanwendungen. Der Funktionsumfang von Actionscript 3.0 ist im Sinne der Erstellung grafischer Benutzeroberflächen im Vergleich zu PHP und anderen Skriptsprachen um einiges weiter entwickelt. Das Erstellen von Vektorgrafiken, das Anwenden aller in Photoshop⁶⁰ möglichen Filtertechniken, die von Grafikprogrammen genutzte Maskentechnik und viele verschiedene Animationstechniken sind nur einige Funktionen, durch welche sich Actionscript auszeichnet. Zu dem basiert Actionscript 3.0 auf einem objektorientierten Ansatz, welcher das Entwickeln einer Software nach modernen Programmierstandards ermöglicht.

Das SWF-Format und das damit verbundene Sicherheitskonzept

Das SWF-Format (Shockwave Flash) ist ein von Adobe eigens konzipiertes Dateiformat. Die mit Adobe Flash erstellten Quelldateien (FLA-Dateien) werden in SWF-Dateien kompiliert und dabei auf Wunsch komprimiert, um anschließend auf einen Webserver übertragen zu werden. Die SWF-Datei ist direkt in einem Webbrowser ausführbar. Vorausgesetzt wird ein installiertes Flashplayer-Plugin. Alternativ können SWF-Dateien clientseitig gespeichert und mit Hilfe eines Flash Players gestartet werden. Da nicht jeder Arbeitsplatzrechner über einen installierten Flashplayer verfügt, ist es möglich Flash-Quelldateien auch als ausführbare Da-

⁶⁰ Adobe Photoshop ist ein Programm zur Bearbeitung pixelorientierter Grafiken. Im Bereich der professionellen Bildbearbeitung ist das Programm Marktführer. Stand Mai 2010

tei (EXE) zu erstellen. Diese Dateien sind um einiges größer als das SWF-Format, da in ihnen der Flash Player zusätzlich integriert ist.

Beim Ausführen von SWF-Dateien wird ein bestimmtes Sicherheitskonzept verfolgt. Das Kopieren, Speichern und Löschen von Daten auf dem Arbeitsplatzrechner ist seitens Flash prinzipiell nicht möglich. Der Nutzer kann grundsätzlich davon ausgehen, dass eine Flash-Anwendung keinen Schaden auf dem eigenen Arbeitsplatzrechner verursachen kann.

Dieses Prinzip wird als Sandboxverfahren⁶¹ bezeichnet. Eine SWF-Datei aus einer bestimmten Internetdomäne kann nur auf Daten (bspw. andere SWF-, Bitmap-, Audio- oder Textdateien) der gleichen Domäne zugreifen. Diese Daten werden in einer Sicherheitsgruppe abgelegt, die auch als Sicherheits-Sandbox bezeichnet wird. Darüber hinaus ist jederzeit ein Cross-Scripting zwischen SWF-Dateien aus der gleichen Domäne zulässig, sofern beide Dateien mit ActionScript 3.0 geschrieben wurden. Als Cross-Scripting wird die Möglichkeit bezeichnet, dass eine SWF-Datei mittels ActionScript auf die Eigenschaften, Methoden und Objekte einer anderen SWF-Datei zugreifen kann. Der Zugriff auf SWF-Dateien anderer Domains ist grundsätzlich untersagt, kann jedoch durch einen Aufruf der Methode `security.allowDomain()` in der geladenen SWF-Datei gezielt ermöglicht werden.

Actionscript 3.0, Objektorientiertes Programmieren und die Actionscript 3.0 Funktionsbibliothek

Die in Flash CS4 eingebettete Programmiersprache wird mit Actionscript 3.0 bezeichnet. Sie basiert auf dem ECMAScript-Standard, welcher unter anderem auch von Javascript genutzt wird. Ist ein Entwickler bereits mit derartigen Programmiersprachen vertraut, wird es ihm nicht schwer fallen mit Actionscript zu programmieren. Seit Actionscript 2.0 unterstützt Adobe das objektorientierte Entwickeln von Software.

Objektorientierte Programmierung (OOP) ist eine Möglichkeit, den Programmcode durch das Zusammenfassen in Objekte möglichst übersichtlich zu strukturieren. Objekte umfassen einzelne Elemente, die Informationen (Datenwerte) und bestimmte Funktionsmerkmale (Methoden). Das Bündeln dieser Werte und Funktionen bringt verschiedene Vorteile. Beispielsweise muss nur eine statt mehrerer Variablen verfolgt, verwandte Funktionen können gemeinsam strukturiert, und es können Programme so aufgebaut werden, dass sie der realen Welt besser entsprechen und damit leichter zu verstehen sind.

⁶¹ Vgl. [hlpadobeProg2010]

Damit nicht jedes Objekt bzw. die Funktionen einer Anwendung neu erstellt werden müssen, bietet Flash eine eigene Komponenten-Bibliothek. In dieser befinden sich zahlreiche Pakete, Klassen, Funktionen, Ereignisse und Fehlermeldungen zur Arbeit mit Actionscript⁶². Für die Entwicklung des Personalinformationssystems sind drei Pakete von besonderer Bedeutung: das Paket der sogenannte obersten Ebene, das flash.*-Paket und Komponenten des fl.*-Pakets. Die oberste Ebene enthält Methoden, wie Date, Math und XML, oder den Sprach-elementen, die auf dem ECMAScript basieren. Unter den flash.*-Paketen befinden sich Klassen der Flash Player API. Mit diesen Klassen können unter anderem Vektorgrafiken (flash.geom.*) erstellt, Photoshop-Techniken angewendet (flash.filters.*) oder Funktionen für komplexere Internet-Anwendungen (z.B. das flash.net-Paket zur Verarbeitung von Datenübertragungen) genutzt werden. Die Klassen des fl.*-Pakets sollen den Funktionsumfang von Actionscript zusätzlich erweitern. Dazu gehören unter anderem: Die ActionScript 3.0 Tween- und Transition-Klassen (fl.transitions) zur Erzeugung von Bewegungsabläufen, eine Locale-Klasse für mehrsprachigen Text (fl.lang) oder die FLVPlayback-Komponentenklassen (fl.video).

One-Frame-Applikation mit Actionscript oder zeitleistenbasierende Animation mit der Flash Bühne

Die Flashbühne ist das zentrale Element des Flash-Bildschirms. Hier können Grafiken erstellt und in Kombination mit einer Zeitleiste zu Animationen verarbeitet werden. Die Steuerung dieser erfolgt durch in einzelnen Frames oder in verschiedenen Ebenen hinterlegtem Actionscript-Code. Die Grafiken sind vektorbasierend und mit Hilfe des Werkzeugtools von Flash schnell und leicht zu erstellen. Diese Entwicklungstechnik wird oft verwendet, hat jedoch einige Nachteile bei der Erstellung eines Personalinformationssystems. Auf Grund der gehobenen Projektgröße gestaltet sich der programmiertechnische Entwicklungsprozess zu unübersichtlich. Bei einer zu großen Anzahl an Frames und Ebenen können einzelne Elemente leicht verloren gehen und dadurch unangenehme schwer auffindbare Fehler verursachen. Zusätzlich wird bei der Erstellung und dem Wiederverwenden komplizierter Grafiken mehr Speicherplatz als bei sogenannten One-Frame-Applikationen benötigt.

One-Frame-Applikationen sind von der Zeitleiste unabhängige vollständig mit Actionscript umgesetzte Anwendungen. Grafiken werden ausschließlich mit Quellcode umgesetzt und durch Flash kompiliert. Nachteil ist, dass die grafische Benutzeroberfläche nur beim Ausführen und nicht während des Entwicklungsprozesses angezeigt wird. Die angelegten Objekte

⁶² Die Übersicht aller von Flash angebotenen Actionscript-Klassen befindet sich unter: **[hlpadobePS2010]**

erfordern anfänglich einen minimal größeren Speicherbedarf als die mit der Flash-Bühne erstellten Elemente. Auf Grund der Wiederverwendbarkeit der einzelnen Methoden und deren Anpassungsfähigkeit durch die Angabe von Parametern wird jedoch schnell eine Speicherreduzierung erreicht. Ein sich im Anhang befindender Vergleichstest unterstützt diese These. Um den client-serverseitigen Übertragungsprozess eines Personalinformationssystems zu optimieren, sollten ausschließlich quillcodebasierende Objekte entwickelt werden.

Der Bandbreitenprofiler

Mit dem in Flash angebotenen Bandbreitenprofiler kann der Downloadvorgang einer SWF und der durch diese verursachten Ladenvorgänge in der Entwicklungsumgebung simuliert und grafisch dargestellt werden. Es wird angezeigt, welche Datenmenge je nach der angegebenen Modemgeschwindigkeit gesendet und wie viel Zeit bei der Übertragung dieser benötigt wird. Der Bandbreitenprofiler ist ein wichtigstes Tool um die Effizienz der Anwendung zu testen und wird bei zahlreichen Vergleichstests zur Wahl einer für die Entwicklung des Personalinformationssystems optimalen Technik eingesetzt.

4 Konzeption eines Personalinformationssystems

Ein wesentlicher Bestandteil des Entwicklungsprozesses ist die Konzeption der Software. Eigene Erfahrungen haben gezeigt, dass gerade bei der Zusammenarbeit mit mehreren Entwicklergruppen vermeidbare Fehler durch schlechte oder fehlende Absprachen entstehen. Auf Grund falscher Lösungsansätze führt dies im schlimmsten Fall zu einer Neuentwicklung der Anwendung. Ein einheitliches und für jede Entwicklerpartei verständliches Konzept ist daher notwendig. Je komplexer die Software wird, umso wichtiger ist eine gründliche Konzeption.

Im folgenden Kapitel wird unter Berücksichtigung der vorangegangenen Kapitel anhand einer allgemeinen Beispielanwendung das Konzept eines Personalinformationssystems beschrieben. Es wird ein allgemeines Grobkonzept erstellt und dessen Teilkomponenten detailliert dargestellt. Das hier vorgestellte Konzept enthält die wichtigsten Ansätze einer solchen Anwendung, kann aber einsatzspezifisch durch weitere Applikationen ergänzt werden.

In der Praxis werden verschiedene Konzeptionstechniken eingesetzt. Eigens gewählte Entwurfsmuster dienen der Darstellung der grafischen Benutzeroberfläche und der konzeptionellen Anordnung der darin enthaltenen Komponenten. Zur Beschreibung der vorhandenen Programmabläufe werden größtenteils⁶³ sogenannte UML-Diagramme verwendet. Diese können allgemein für die übersichtliche Darstellung der Makroarchitektur oder detailliert mit programmrelevanten Parametern im Sinne der Mikroarchitektur erstellt werden.

Um einen übersichtlichen Rahmen der konzeptionellen Aufgaben in dieser Arbeit beizubehalten, werden die einzelnen Konzeptionstechniken abwechselnd aufgezeigt. Nicht dargestellte Teilbereiche sind zusätzlich im Anhang zu finden.

4.1 Das Grobkonzept der Beispielanwendung

In den einführenden Kapiteln dieser Arbeit wurde darüber berichtet, dass Personalinformationssysteme drei Hauptaufgaben des Personalwesens unterstützen. Sie versorgen den Nutzer mit Mitarbeiterinformationen, steuern betriebliche Abläufe und helfen bei der Kontrolle vorhandener Arbeitsprozesse. Zur Umsetzung dieser Anforderungen werden in der Beispielanwendung die drei Bereiche „Personal“, „Steuerprozesse“ und „Kalender“ angelegt. Alle

⁶³ Alternativ zu UML-Diagrammen können auch andere Techniken, bspw. das „Entity Relationship Model“, verwendet werden.

weiteren hier dargestellten Programmteile dienen zur Ausführung und Organisation der Anwendung selbst. In der Beispielanwendung werden zu diesem Zweck die Module „Login-Bereich“ und „Operator“ angelegt. Der Login-Bereich unterstützt das Authentifizieren des Nutzers und das Laden programmrelevanter Daten. Durch den Operatorbereich wird das Anlegen der für das PIS notwendigen Hintergrundinformationen ermöglicht.

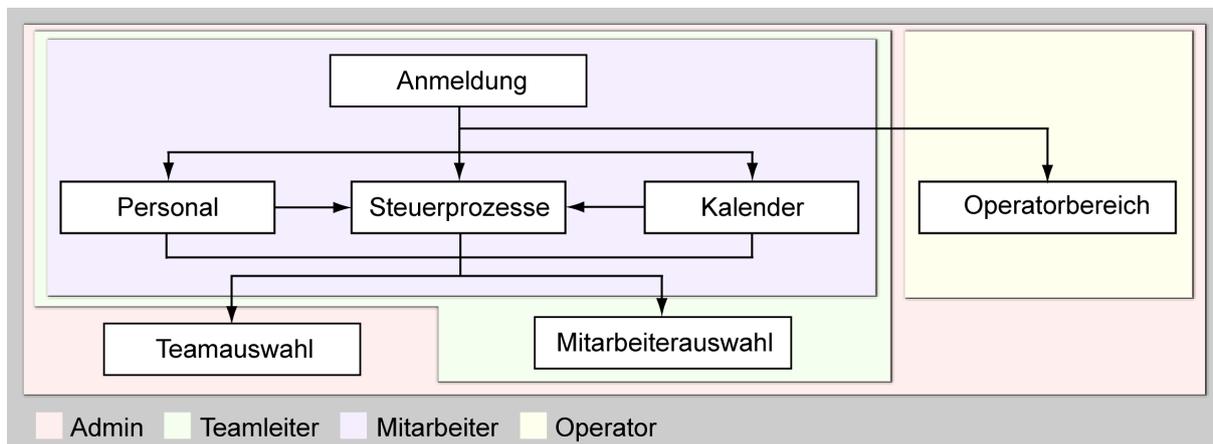


Abbildung 13: Grobkonzept der Beispielanwendung

Der Login-Bereich

Vor dem eigentlichen Zugriff auf die Kernprozesse eines Personalinformationssystems müssen die Nutzerrechte der einzelnen Programmteile und -inhalte einer Person geprüft werden. Erst bei erfolgreicher Anmeldung werden die für die Arbeit auf dem Arbeitsplatzrechner notwendigen Daten geladen und freigeschaltet. Der Login-Bereich realisiert diesen Authentifizierungs- und Ladeprozess.

Der Personal-Bereich

Der Bereich Personal beinhaltet die Darstellung persönlicher und firmengebundener Mitarbeiterinformationen, einen Bewertungsbereich der Mitarbeiterkompetenzen und ein Verwaltungsmodul für die eben genannten Informationen. Das Ändern dieser Inhalte kann Auswirkung auf die Steuerprozesse oder die Kalenderfunktionen haben. So werden beispielsweise Daten im Bereich der Personalkompetenzen durch die Steuerprozesse aktualisiert und sollten nur im Falle falsch angelegter Arbeitsabläufe von Hand korrigiert werden. Zur Umsetzung dieser Module werden unter Personal die Teilbereich „Personalinformation“, „Personalkompetenzen“ und „Personalverwaltung“ angelegt.

Der Bereich der Steuerprozesse

Der Teilbereich der Steuerprozesse dient der Planung und Koordinierung der im Betrieb vorhandenen Arbeitsabläufe. Unter Berücksichtigung der Personalkompetenzen und der Arbeitsplatzinformationen können Dienstpläne oder Pläne vorhandener Raumkapazitäten erstellt werden. Diese werden ausgewertet und die daraus resultierenden Ergebnisse bei vollständig ausgeführten Arbeitsprozess auf die Personalkompetenzen projiziert.

Der Kalenderbereich

Der Kalenderbereich dient der mittel- und längerfristigen Planung betrieblicher Prozesse. Mitarbeiter können Termine, wie Urlaub, Abbau von Überstunden oder Weiterbildungsmaßnahmen, beantragen und durch eine verantwortliche Führungsperson bestätigen lassen. Wird ein Termin zugelassen, wird der Mitarbeiter für den entsprechenden Zeitraum von den Steuerprozessen ausgeschlossen.

Der Operatorbereich

Zum Ausführen der Steuerprozesse und der damit verbundenen Personalbewertung ist das Anlegen von Hintergrundinformationen auf die vorhandenen Arbeitsplatzressourcen notwendig. Der Operatorbereich ermöglicht und unterstützt diese Aufgabe. Zugriff auf diesen Bereich haben in der Regel nur die sogenannten Operators und der Admin.

4.2 Konzeption des Login-Bereichs mit Preloading-Prozess

Bevor der Nutzer mit der Anwendung arbeiten darf, gilt es einen Authentifizierungsprozess durchzuführen. Dieser prüft, ob und welche Zugriffsrechte einer Person zugeteilt sind. Bei erfolgreicher Anmeldung wird ein sogenannter Preloading-Prozess gestartet, welcher die für die clientseitige Arbeit notwendigen Anwendungsdaten auf den Arbeitsplatzrechner lädt und vorbereitet.

4.2.1 Der Login-Bereich

Zum Anmelden eines Nutzers gibt dieser Benutzername und -passwort in eine Anmelde-
maske ein und sendet diese an den Server. Anhand einer dort hinterlegten Datenbank wird

die Korrektheit der Nutzerdaten geprüft. Eine Nutzer-ID und ein Nutzerstatus werden bei erfolgreichem Login an den Client zurück gesendet und das Initialisieren der Software begonnen. Missglückt der Anmeldeprozess werden alle Eingabefelder zurückgesetzt und eine entsprechende Fehlermeldung angezeigt. Die Nutzerrechte werden beim Anlegen einer neuen Person von einem Programmadministrator vergeben. Die Beispielanwendung unterstützt prinzipiell vier Nutzergruppen: den Admin, den Teamleiter, den Mitarbeiter und den Operator.

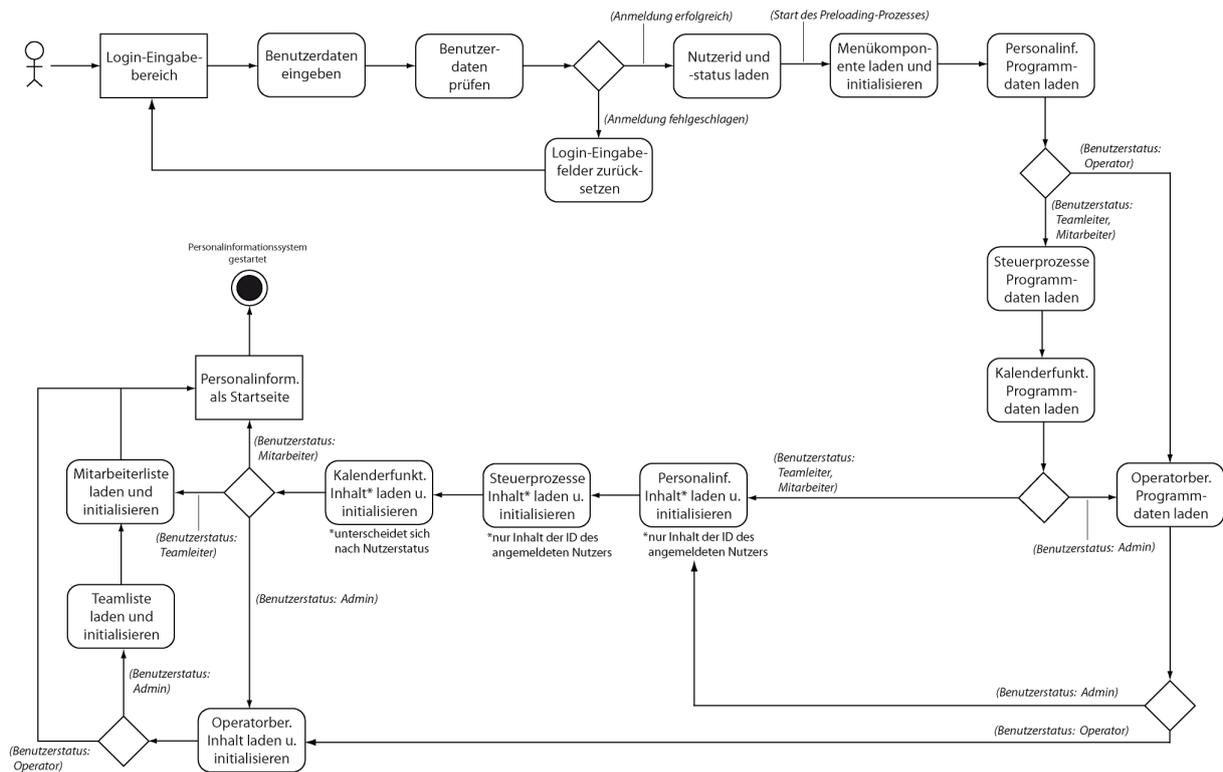


Abbildung 14: UML des Anmelde- und Preloading-Prozesses

4.2.2 Der Preloading-Prozess

Ist die Anmeldung vollzogen, werden die für die clientseitige Arbeit notwendigen Programmteile auf den Arbeitsplatz übertragen und mit entsprechendem Inhalt initialisiert. Abbildung 13 zeigt einen solchen Vorgang. Die Programmmodule sind in separaten Dateien hinterlegt und werden nur bei erteilten Zugriffrechten geladen. Programmdateien und -inhalte ohne Zugriffrechte werden von dem Preloading-Prozess ausgeschlossen. Mit geringer werdender Zahl an Nutzerechten werden so eine enorme Verkürzung des Ladeprozesses und ein zusätzlicher Sicherheitsgewinn erreicht.

Ist der Preloading-Prozess abgeschlossen, wird das Programm in der Beispielanwendung mit dem Menüpunkt der Personalinformation gestartet. Alternativ könnten an dieser Stelle

andere Menüpunkte oder eine funktionsunabhängige Startseite gewählt werden.

4.3 Konzeption des Personalbereichs

Der Kernprozess Personal beinhaltet alle notwendigen Aufgaben bezüglich der Darstellung und Verwaltung von Informationen der in einem Betrieb angestellten Mitarbeiter. Unterschieden wird in personengebundene bzw. firmenorganisatorische Angaben und die Personalbewertung als Grundlage betriebsinterner Steuerprozesse.

4.3.1 Darstellung der Personalinformationen

Der Schwerpunkt der Personalinformationen liegt in der Darstellung von personengebundenen und firmenorganisatorischen Daten. Kurze Zugriffszeiten und die übersichtliche Präsentation der Informationen stehen bei der Umsetzung dieses Bereichs im Vordergrund. Lässt die Zahl anzuzeigender Informationen keine übersichtliche Darstellung zu, muss dieser Bereich in weitere Unterkategorien unterteilt werden. Die Bereitstellung der personengebundenen Daten erfolgt nach dem Datenschutzgesetz⁶⁴ und muss mit dem Betriebsrat abgesprochen sein.

Folgende Angaben können unter anderem im Bereich Personalinformationen eingesetzt werden:

Persönliche Daten:	Name, Vorname, Adresse, Alter, Titel, Passbild, Führerscheinklasse
Firmenorganisatorische Daten:	Büronummer, Abteilungsname, Firmentelefon, Eintrittsdatum, Kostenstelle, Team, Frist des Arbeitsvertrages

Der Zugriff auf die Personaldaten einer Person erfolgt entweder über eine Team- bzw. Mitarbeiterauswahl oder über eine Suchfunktion. Der Inhalt der einzelnen Elemente wird personengebunden aus einer Datenbank geladen. Die zu ladenden Informationen sind eindeutig mit einer Personen-ID verknüpft und werden direkt in eine Darstellungsmaske integriert.

⁶⁴ Vgl. [PaerliCaplaziSuter2007], 128

4.3.2 Darstellung der Personalkompetenzen

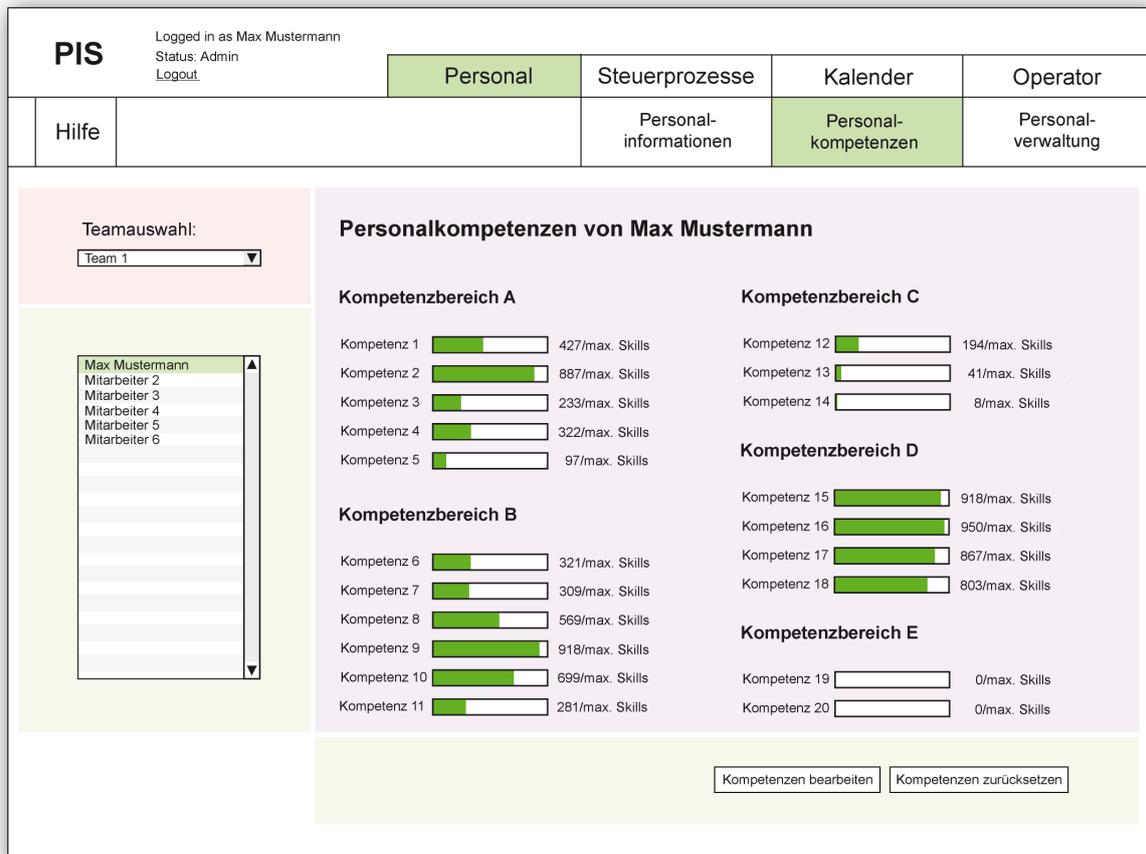


Abbildung 15: Entwurfsmuster Personalkompetenzen

Der Bereich der Personalkompetenzen bietet dem Nutzer eine Übersicht aller in den Steuerprozessen verwendeten und einem Mitarbeiter zugewiesenen Bewertungskriterien. Welche Kompetenzen angelegt werden, ist von dem jeweiligen Betrieb und den dort vorhandenen Arbeitsplatzressourcen abhängig. Die übersichtliche Darstellung und kurze Zugriffswege stehen auch in diesem Bereich im Vordergrund. Um dies zu gewährleisten, sind die einzelnen Eigenschaften verschiedenen Kategorien zugeordnet. Die Mitarbeiterauswahl erfolgt kongruent zur Personalinformation über eine Team- bzw. Mitarbeiterauswahl oder über eine Suchfunktion. Die Informationen selbst werden in numerischer Form personengebunden in der Datenbank gespeichert und beim Laden mittels eines Referenzwertes verrechnet. Der so entstandene Wert wird als Parameter zur Erzeugung der in der Personalkompetenz verwendeten Komponenten genutzt. Dabei existieren die verschiedensten Darstellungsformen. So können die einzelnen Anzeigeobjekte als numerischer Wert, als einem durch einen Wertebereich zugeordneten Fremdbezeichner oder, wie in Abbildung 15 in statistischer Form (hier Balkendiagramme) dargestellt werden.

4.3.3 Verwaltung der Personalinformationen und -kompetenzen

Damit die Personendaten erstellt, bearbeitet oder gelöscht werden können, wird in der Beispielanwendung das Programmmodul „Verwaltung der Personalinformationen“ angelegt. Diese Softwarekomponente wird in einer separaten Datei gespeichert und nur dem Teamleiter und dem Admin zur Verfügung gestellt. Die darin enthaltenen Einzelkomponenten werden mit den personengebundenen Informationen initialisiert und ermöglichen das Editieren der Komponenteninhalte. Ist der Verarbeitungsprozess abgeschlossen wird der bereits vorhandene Datensatz in der Datenbank überschrieben. Die Struktur der Anwenderoberfläche und die Datenbasis sind mit den Bereichen der Personaldatendarstellung identisch.

Das Anlegen der Daten erfolgt manuell auf der Grundlage üblicher Personalbögen⁶⁵. Dieser Prozess ist sehr aufwendig und soll mittels verschiedener Eingabefunktionen des Personalinformationssystem erleichtert werden.

Unterstützung durch Personalprofile

In vielen Unternehmen werden Mitarbeiter unter einer ihnen zugeteilten Personalbezeichnung geführt. Innerhalb dieser Zuordnung werden den Angestellten identische Werte einzelner Personalangaben übertragen. Um ein sich wiederholendes Eintragen dieser Information zu umgehen, hat der Nutzer die Möglichkeit sogenannte Personalprofile zu laden. Zu diesem Zweck werden einmalig Standardwerte in eine Personalmaske eingetragen und diese unter einem frei wählbaren Namen gespeichert. Beim Laden eines Personalprofils werden diese Standardwerte in die Programmkomponenten der ausgewählten Person eingefügt und zur weiteren Bearbeitung freigegeben.

Halbautomatisierte Informationen

In einzelnen Fällen kann der Informationsgehalt einer Informationskomponente durch den Bezug auf ein anderes Datenobjekt abgeleitet werden. So berechnet sich zum Beispiel mittels Geburtsdatum und dem allgemeinem Renteneintrittsalter der voraussichtliche Firmenaustritt einer Person. Informationsmodule dieser Art sind einer oder mehreren Referenzkomponenten und einer Berechnungsfunktion zugeordnet. Der so erzeugte Wert kann auf Wunsch mit der Initialisierung des Referenzobjekts automatisch übernommen werden.

⁶⁵ Vgl. [Zapp2006], 63

Betriebsbedingte Informationen in Verbindung mit HW-Systemen

Ein weiteres Mittel für die automatisierte Informationserfassung, ist die Verknüpfung des PIS mit zusätzlichen Hardwaresystemen. Zeiterfassungssysteme verschiedenster Anbieter werden gerade bei gleitenden Arbeitszeiten von vielen Betrieben eingesetzt. Die Implementierung einer entsprechenden Schnittstelle zu diesen Geräten kann unter anderem im Bereich der Gehaltsabrechnung und der Arbeitszeiterfassung zu einer starken Verringerung der personalverwaltenden Aufgaben führen. Da die weitere Ausführung dieses Themas zur Überschreitung des in dieser Arbeit vorgegebenen Rahmens führt, wird die Informationserfassung mittels zusätzlicher Hardware nicht im Konzept der Beispielanwendung berücksichtigt.

Automatisierte Informationen

Die vollautomatisierte Erstellung von Informationen findet sich hauptsächlich im Bereich der Personalkompetenzen wieder. Während des Ablaufs der betrieblichen Steuerprozesse aktualisiert das Personalinformationssystem automatisch die Werte eines Mitarbeiters. Das Editieren und Verwalten dieser Informationen erfolgt in der Regel nur zur Erstellung einer grundlegenden Datenbasis und bei fehlerhaft angelegten Arbeitsabläufen.

4.4 Konzeption der Steuerprozesse

Die Koordination der betrieblichen Abläufe eines Unternehmens ist zeit- und kostenaufwendig. Die Dienstplanerstellung und die Zuordnung der Mitarbeiter auf die verschiedenen Arbeitsplatzressourcen regeln in den meisten Betrieben die Erfahrungen und das firmenspezifische Wissen der für einen Mitarbeiterstamm verantwortlichen Führungspersonen. Die Programmfunktionen des Personalinformationssystems sollen diese Aufgaben unterstützen, wenn nicht sogar vollständig übernehmen.

Die Steuerprozesse der Personalverteilung und der Raumplanung werden im folgenden Kapitel näher erläutert, da sie in ihrer Allgemeinheit in vielen Unternehmen zum Einsatz kommen. Die betriebsbedingten Strukturen können in dieser Arbeit nicht berücksichtigt und müssten in der Anwendung zusätzlich an das jeweilige Unternehmen angepasst werden.

4.4.1 Die Personalverteilung (Dienstplanerstellung)

Das optimale Verteilen eines Personenstammes auf die vorhandenen Arbeitsplatzressourcen ist ein mit steigender Anzahl zu betreuender Mitarbeiter wachsender Arbeitsprozess. Bisher verwendete Techniken, wie das Verwenden von Excel-Tabellen, unterstützen zwar diese Arbeit, können aber nur in Verbindung mit den Erfahrungen, dem Wissen und dem Handeln verantwortlicher Führungsperson genutzt werden. Um den Prozess der Personenverteilung zusätzlich zu unterstützen, wird der Informationsgehalt eines Teamleiters erfasst und für automatisierte Steuerroutrinen verarbeitet.

Die folgenden Kriterien sind für die Verteilung von Personen grundsätzlich notwendig: die Personenkompetenz, die Personenverfügbarkeit, arbeitsplatzspezifische Eigenschaften und die Arbeitsplatzverfügbarkeit. Diese Faktoren werden auf ein bestimmtes Datum oder ein Zeitintervall projiziert und unter einem ausgewählten Verteilungsschlüssel verarbeitet. Die für die Verteilung zuständigen Personen können anhand dieser Informationen das Personalinformationssystem mittels verschiedener Automatisierungsstufen unterstützend einsetzen.

Halbautomatisches System - Teamleiter und Admin

Die einfachste Vorgehensweise der Dienstplanerstellung erfolgt über die einzelne Zuordnung des Personals auf die Arbeitsplatzressourcen durch den Teamleiter selbst. Dieser wählt ein Datum und einen Arbeitsbereich aus. In diesem werden ihm alle zur Verfügung stehenden Arbeitsplatzressourcen übersichtlich dargestellt. Er hat nun die Möglichkeit aus einer Mitarbeiterliste einzelne Personen auszuwählen und diese dem von ihm als passend erachteten Arbeitsplatz zuzuordnen. Daraus folgend werden die Mitarbeiter aus der Liste entfernt und der Arbeitsplatz als besetzt gekennzeichnet. Zur Unterstützung dieses Vorgangs werden in der Raumdarstellung die wichtigsten Informationen der einzelnen Arbeitsplatzressourcen und die Mitarbeiterkompetenzen dargestellt. Das Hinzuziehen detaillierte Personeninformationen, wie beispielweise bereits gesetzte Termine eines Mitarbeiters in der laufenden Woche oder Sortierfunktionen der Mitarbeiterliste, ermöglicht eine weitere Unterstützung des Verteilungsprozesses.

Halbautomatisches System - Mitarbeiter

In einigen Unternehmen wird die Dienstplanerstellung durch die Mitarbeiter durchgeführt. Der einzelne Angestellte trägt sich selbständig zu einem bestimmten Datum in die noch nicht besetzten Arbeitsplatzressourcen ein. Das Personalinformationssystem stellt ihm eine in der

Darstellung begrenzte Raumübersicht zur Verfügung. Abgebildet werden nur die Arbeitsplatzinformationen und die Arbeitsplatzverfügbarkeit. Das Anzeigen anderer bereits eingetragener Kollegen ist nur mit der Zustimmung des Betriebsrates oder der zuständigen Führungsetage möglich. Mit der Wahl eines Arbeitsplatzes, wird dieser für anderen Mitarbeiter als besetzt gekennzeichnet.

Der Teamleiter und der Admin übernehmen an dieser Stelle nur eine kontrollierende Funktion. Das PIS ermöglicht ihnen, dass ständige Bearbeiten hinterlegter Daten. Bereits laufende oder zurückliegende Prozesse sind nicht veränderbar. Bei einer entsprechenden Änderung wird der betroffene Mitarbeiter in Kenntnis gesetzt.

Das Speichern und Laden von Dienstplänen

In vielen Unternehmen existiert ein sich periodisch wiederholender Dienstplan. Mitarbeiter sind zu bestimmten Zeiten immer dem gleichen Arbeitsplatz zugeordnet. Um eine sich ständig wiederholende Eingabe zu vermeiden, können Dienstpläne für ein bestimmtes Zeitintervall gespeichert und später in einen anderen Zeitraum gleicher Länge aufgerufen werden. Steht ein Mitarbeiter für das neu gewählte Zeitintervall nicht zur Verfügung, wird das in der Raumdarstellung besonders gekennzeichnet und der Angestellte von den Berechnungsprozess der Personenkompetenzen ausgeschlossen.

Vom halbautomatisierten zum vollautomatisierten System

Die Verteilung von Personen erfolgt durch eine vom Unternehmen bzw. den verantwortlichen Führungspersonen festgelegte Vorgehensweise. Es gibt immer einen eindeutig definierbaren Grund, warum der Mitarbeiters A auf den Arbeitsplatz X gesetzt wurde. Die funktionelle Erfassung dieser Zuordnungsmethoden in Verbindung mit den Personen- und Arbeitsplatzdaten automatisiert die Erstellung eines Dienstplanes. Bei derartigen Zuteilungsprozessen sind unter anderem die folgenden Verteilungsszenarien möglich:

Höchste/Geringste Kompetenz auf den zugehörigen Arbeitsplatz: Dieser Prozess vergleicht die Mitarbeiterkompetenzen mit Arbeitsplatzanforderungen und verteilt die Personen möglichst optimal nach Ihren Kompetenzen.

Gleichverteilung nach Nutzungsdauer der Arbeitsplatzressourcen: Mitarbeiter sollen möglichst die gleiche Anzahl an Arbeitsstunden an den zur Verfügung stehenden Arbeitsplatzressourcen leisten.

Trotz Zuhilfenahme des Verteilungsschlüssels wird diese Methode weiterhin als halbautomatisches System beschrieben, da Erstellung des Dienstplanes nur durch aktives Handeln des Teamleiters zu Stande kommt. Dieser wählt einen entsprechenden Zeitraum mit zugehörigem Startdatum und einen von ihm gewünschten Verteilungsschlüssel. Das System erkennt die Eigenschaften gewählter Personen, vergleicht diese mit den Arbeitsplatzressourcen und vollzieht anschließend anhand der zugehörigen Daten einen optimierten Verteilungsprozess. Eine weitere Verfeinerung des Verteilungsschlüssels kann die Zuordnungsmethode durch das Ergänzen von zusätzlichen Angaben erhalten. So sollte es beispielsweise möglich sein einen Mitarbeiter aus dem Verteilungsprozess herauszulösen und diesen fest an einen ihm zugeteilten Arbeitsplatz zu binden. Die Berechnung des Verteilungsprozesses erfolgt dann ohne diese Dienstplanressourcen.

Vollautomatische Systeme

Bei vollautomatischen Systemen wählt der Nutzer einmalig einen Verteilungsschlüssel. Dieser wird durch ein Berechnungs- und Erstellungsintervall ergänzt. Alle weiteren Aufgaben übernimmt das Personalinformationssystem. Das Erstellungsintervall gibt den regelmäßigen Zeitraum vor, für welchen ein Dienstplan erstellt wird. Der Umgang mit den Personalkompetenzen spielt dabei eine entscheidende Rolle. Es gilt zu entscheiden, ob die aktuellen Kompetenzen dauerhaft für den entsprechenden Zeitraum oder die durch den laufenden Prozess entstehenden Beurteilungen zu nutzen sind. Das Berechnungsintervall ist die Zeitspanne, nach welcher eine Aktualisierung der Personalkompetenzen durchgeführt wird. Im Normalfall ist das ein Run⁶⁶, eine Arbeitsschicht oder ein Arbeitstag. Der Aktualisierungsprozess ist bei Bedarf auch rückwirkend auf einen längeren Zeitraum anwendbar.

Als zusätzliche Komponente bezieht das Personalinformationssystem Daten der mittleren- und längerfristigen Planung in den Berechnungsvorgang ein. Ist ein Mitarbeiter in einem bestimmten Zeitraum nicht verfügbar, wird dies vom System erkannt und die fehlende Dienstplanressource von dem Berechnungsprozess ausgeschlossen.

Der Teamleiter bzw. der Admin übernimmt bei automatisierten Abläufen lediglich eine prüfende Funktion. Er kann weiterhin Änderungen am Dienstplan vornehmen und eine automatische Neuberechnung zukünftiger Arbeitstage anordnen.

⁶⁶ Vgl. Kapitel 4.6.2 Hintergrundinformationen und Runs

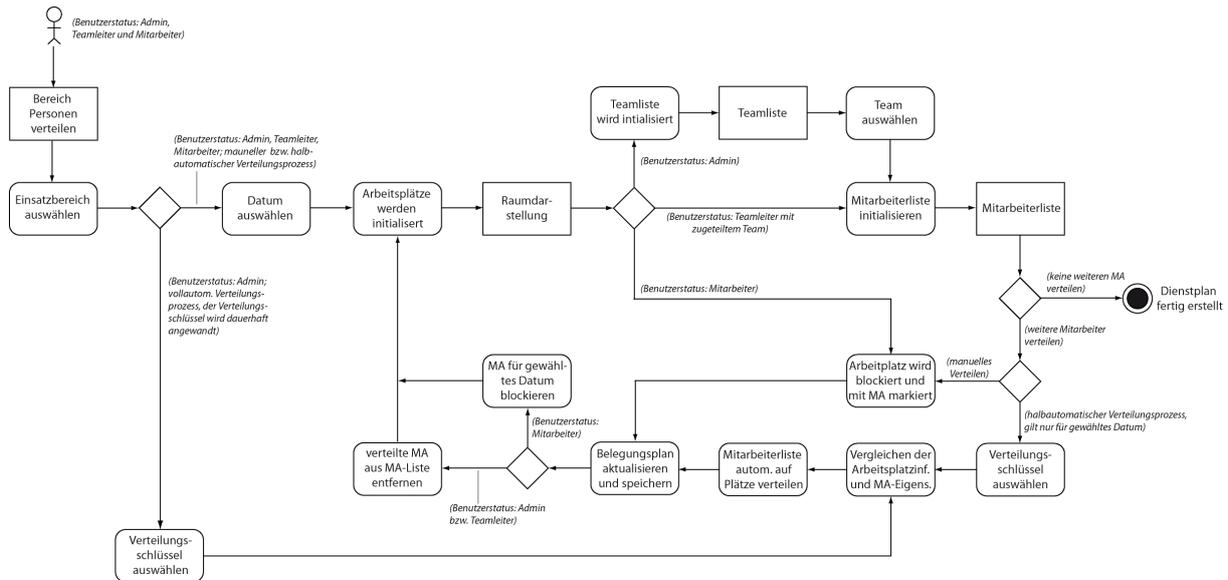


Abbildung 16: Entwurfsmuster Steuerprozesse - Personen verteilen

4.4.2 Raumplanung

Bei einer hohen Anzahl verschiedener Räumlichkeiten wird in vielen Betrieben oder Instituten durch den damit verbundenen Zuteilungsprozess ein großer organisatorischer Aufwand erzeugt. Räume werden ähnlich der Dienstplanerstellung längerfristig an Personen bzw. Personengruppen gebunden oder kurzfristig verteilt. Die Raumverteilung kann dauerhaft, ein Mitarbeiter wird dabei einem Arbeitsplatz zeitunabhängig zugeteilt, oder variabel organisiert werden. In diesem Fall wird der Nutzer für ein gewisses Zeitfenster an einen Raum gebunden und dieser danach für alle Benutzergruppe wieder freigegeben. Im Gegensatz zur Personenverteilung sind die Rauminformationen fest gesetzt und haben keinen Einfluss auf die Mitarbeiterkompetenzen. Für die optimale Organisation des Raumverteilungsprozesses kann ein Personalinformationssystem unterstützend eingesetzt werden.

Der Raumplanungsbereich ist in zwei Module unterteilt. Zum einen unterstützt eine Suchmaske die Eingabe notwendiger Raumkriterien und zum anderen zeigt eine Ergebniskomponente in Listenform alle passenden Räume an. Durch einen dritten Bereich, eine Teamauswahl mit Personenliste, wird dem Teamleiter bzw. dem Admin das Auswählen von Mitarbeitern für den Raumverteilungsprozess ermöglicht.

Der Verteilungsprozess erfolgt durch die Wahl einer Person und der für diese relevanten Raumdaten. Bei Nutzern mit Mitarbeiterstatus wird die Raumzuordnung direkt für die angemeldete Person angewendet. Die eingegebenen Kriterien werden auf die vorhandenen

Räume projiziert und eine entsprechende Suchroutine durchgeführt. Räume die den Suchkriterien nicht entsprechen und bereits besetzte Plätze erscheinen nicht in der Ergebnisliste. Der Mitarbeiter kann einen der zur Verfügung stehenden Räume auswählen und diesen als besetzt kennzeichnen.

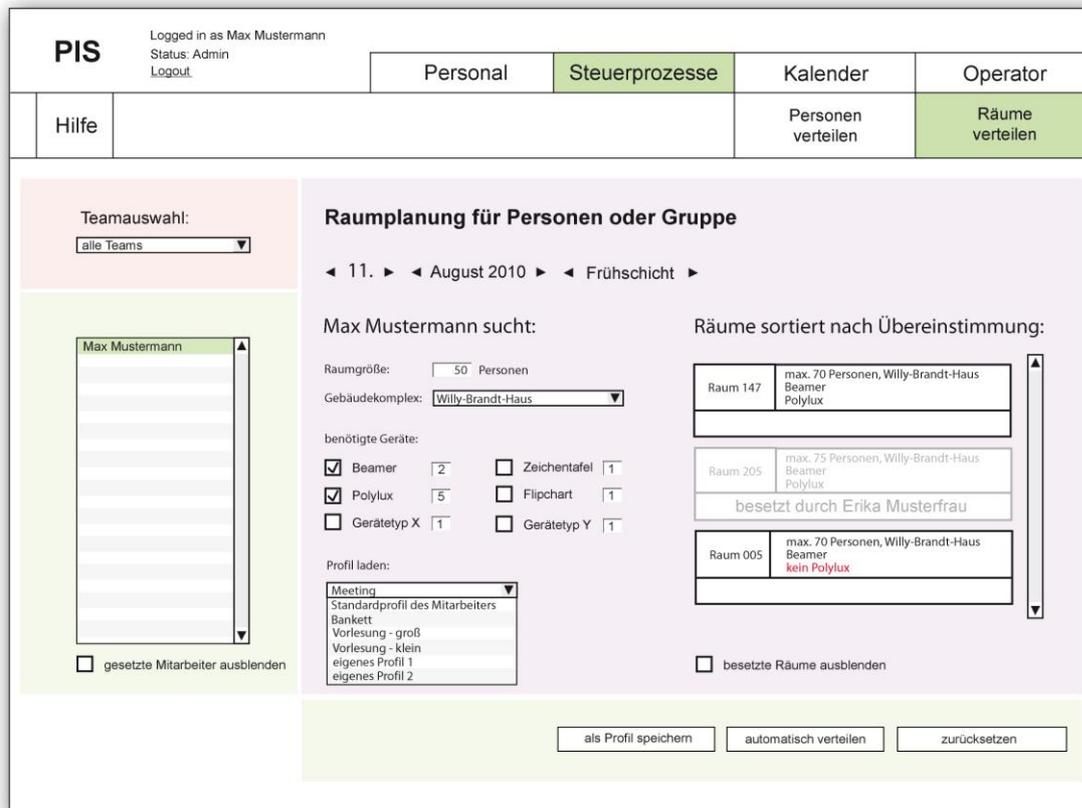


Abbildung 17: Entwurfsmuster für die Raumverteilung

Durch das Darstellen besetzter und unbesetzter Räume und das schnelle Finden dieser durch die Suchfunktion wird der Raumverteilungsprozess erleichtert. Das Zuteilen mehrerer Personen auf verschiedene Arbeitsplätze erfordert jedoch weiterhin einen hohen Arbeitsaufwand.

Profilfunktionen

Bestimmte Anlässe fordern bestimmte Räume. Damit ein sich ständig wiederholender Eingabeprozess umgangen wird, sind die Einträge einer Suchmaske als Raumprofil gespeichert und bei Bedarf abrufbar. Der Nutzer wählt die Raumkriterien, ohne die Hintergrunddaten eines Events wissen zu müssen, aus und lässt sich eine entsprechende Übersicht passender Räume anzeigen.

Das Anlegen von Mitarbeiterinformationen

Ähnlich der Profilfunktion ist für jeden Mitarbeiter eine an ihn angepasste Suchmaske initialisierbar. Das Aufrufen hinterlegter Mitarbeiterwerte ermöglicht eine effektivere Verarbeitung und kann für automatisierte Verteilungsprozesse genutzt werden. Die dafür zulässigen Steuerroutinen sind mit denen der Personalverteilung identisch⁶⁷.

4.5 Konzeption des Kalenderbereichs

Der Kernbereich Kalender dient der Verwaltung mittel- und längerfristiger Termine. Jeder Mitarbeiter kann ein Datum oder einen Zeitraum auswählen und eine begründete Sperrung der betrieblichen Steuerprozesse beantragen. Gleichzeitig werden ihm alle bereits bestätigten Termine und die ihm zugeteilten Steuerprozesse übersichtlich dargestellt. In der Beispielanwendung wird der Kalenderbereich in zwei Menüpunkte aufgeteilt. In die Kalenderübersicht, zur in kalendarischer Form dargestellten Verwaltung seiner Termine, und in die Kalenderinformation, welche alle beantragten, bestätigten und abgelehnten Termine einer zugehörigen Person übersichtlich auflistet.

4.5.1 Kalenderübersicht

Die Kalenderübersicht dient dem Beantragen mittel- und längerfristiger Termine und dem Anzeigen bereits vorhandener Einträge. Eine Standardübersicht in Kalenderform mit einem durch den Nutzer gewählten Anzeigebereich (Wochenansicht, Monatsansicht, Jahresansicht) sorgt für eine übersichtliche Darstellung aller bereits eingetragenen Werte.

Zum Erstellen neuer Termine wählt der Nutzer direkt über den Kalender oder über eine zusätzliche Eingabehilfe einen Zeitraum aus, er beschreibt diesen mit einem Ausfallgrund und sendet den so erstellten Termin zur weiteren Bearbeitung an den zuständigen Teamleiter. Die Wahl des Ausfallgrundes erfolgt über eine im Programm verankerte Auswahlmöglichkeit. Der Ausfallgrund sollte nur im Sonderfall durch eigene Angaben beschrieben werden. Grund dafür ist der technische Zusammenhang zwischen Personalinformationen und Kalenderfunktion. Werden beispielsweise Urlaubstage beantragt, wird dies in der Personalinformation erkannt und mit den noch zur Verfügung stehenden Urlaubstagen abgestimmt. Weitere Aus-

⁶⁷ Vgl. Kapitel 4.4.1 Das Speichern und Laden von Dienstplänen

fallgründe können Krankheit, Weiterbildungsmaßnahmen, Absetzen von Überstunden oder Dienstreisen sein. Ist eine Anfrage vollständig ausgefüllt, prüft das Personalinformationssystem, ob der beantragte Termin möglich ist, und lehnt im Einzelfall den Antrag von vornherein ab. Den Urlaubsantrag betreffend wäre das bei unzureichender Anzahl ausstehender Urlaubstage der Fall.

PIS Logged in as Max Mustermann
Status: Admin
Logout

Personal Steuerprozesse **Kalender** Operator

Hilfe Kalender-information **Kalender-übersicht**

Teamauswahl:
Team 1

Kalenderdaten von Max Mustermann

◀ August 2010 ▶ aktuelles Datum: 02.06.2010

Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	Sonntag
26	27	28	29	30	31	1
■ Urlaub	■ Urlaub	■ Urlaub	■ Urlaub	■ Urlaub		
2	3	4	5	6	7	8
■ A12345	■ A12345	■ A12345	■ A12345	■ A12345		
■ A12346	■ A12346	■ A12346	■ A12346	■ A12346		
9	10	11	12	13	14	15
■ B45678	■ B45678	■ C13579	■ B98765	■ frei		
16	17	18	19	20	21	22
		■ Dienstreise	■ Dienstreise	■ frei		
23	24	25	26	27	28	29
■ C12345	■ C12345					
30	31	1	2	3	4	5

Legende

- Urlaub
- Weiterbildung
- zugeteilter Run
- Krank
- Sonderaufgaben
- frei/Überstunden absetzen
- Dienstreise
- Projekt X

neuer Termin Termin bearbeiten Termin löschen

Abbildung 18: Entwurfsmuster für die Kalenderübersicht eines Teamleiters

Bereits im Kalender eingetragene Informationen werden bei der Antragsstellung ignoriert und bei erfolgreicher Bestätigung ersetzt. Entsprechende Warnhinweise bei der Erstellung eines Termins und eine deutliche Anzeige im Antragsformular des Teamleiters sollen das nicht gewollte Überschreiben vorhandener Termine verhindern. Ist der Antrag zum Senden an den Teamleiter bereit, wird dieser im Kalender als unbestätigter Termin eingetragen und ein Request zur Bearbeitung weitergeleitet.

Im Bereich Kalenderinformation empfängt der Teamleiter die Anfrage. Er muss entscheiden, ob der Termin bestätigt oder abgelehnt wird. Zu seiner Entscheidungsfindung stehen ihm Übersichten des Personals und der Steuerprozesse zur Verfügung. Wird ein Termin bis zu Beginn des betreffenden Zeitraums nicht bearbeitet, fällt dieser aus der Kalenderübersicht

heraus und wird als unbeantworteter Vorgang archiviert. Bei Bestätigung des Antrags übertragen sich die in diesem enthaltenen Informationen auf den Kalender des Mitarbeiters. Erzeugt dieser Vorgang das Überschreiben bereits vorhandener Termine, kann der Teamleiter eine Neuberechnung der Steuerprozesse anordnen. Durch Ablehnen eines Antrags wird der Termin aus der Kalenderübersicht des Mitarbeiters entfernt und eine entsprechende Mitteilung dem Antragssteller zugesandt.

4.5.2 Kalenderinformationen

In der Kalenderinformation empfangen die Teamleiter bzw. der Admin die von den Mitarbeitern zugesandten Antragformulare. Diese werden in chronologischer Form aufgelistet und stehen zur weiteren Verarbeitung bereit. Der einzelne Mitarbeiter sieht diese Ansicht nicht. Für ihn wird dieser Bereich zur übersichtlichen Darstellung seiner erstellten Anträge und deren Status genutzt.

Die Kalenderinformation für den Teamleiter

Die Kalenderinformation eines Teamleiters oder Administrators dient dem Verwalten von Mitarbeiteranträgen. In einer Übersicht bekommt er in chronologisch geordneter Form alle Anträge der Angestellten aufgelistet. Die Zahl anzuzeigender Objekte kann mittels Team- und Mitarbeiterliste eingegrenzt werden. Eine Antragskomponente besteht aus der kompakten Darstellung der wichtigsten Informationen (Mitarbeitername, Datum, Begründung), den für die Bearbeitung zuständigen Steuerkomponenten und einer für den Aufruf detaillierter Daten⁶⁸ verantwortlichen Hilfefunktion. Ist ein Antrag fertig bearbeitet (bestätigt bzw. abgelehnt), wird dieser aus der Listenkomponente entfernt und in einer Archivierungsdatei gespeichert.

Um den Verarbeitungsprozess des Teamleiters zu erleichtern, werden in der Personalinformation zusätzlich statistische Übersichten dargestellt. Diese Angaben beinhalten unter anderem die Zahl der bereits in den Dienstplan integrierten Personen, die Auslastung der Arbeitsplatzressourcen, die Anzahl der Arbeitsstunden eines Mitarbeiters bzw. einer Personengruppe oder die Anzahl bereits stattgefundener Fehlzeiten. Für eine an den Antrag angepasste Darstellung dieser Informationen können diese mittels Team- und Mitarbeiterliste auf eine Person oder Personengruppe eingegrenzt werden.

⁶⁸ zum Beispiel das Vorhandensein bereits eingetragener Termine

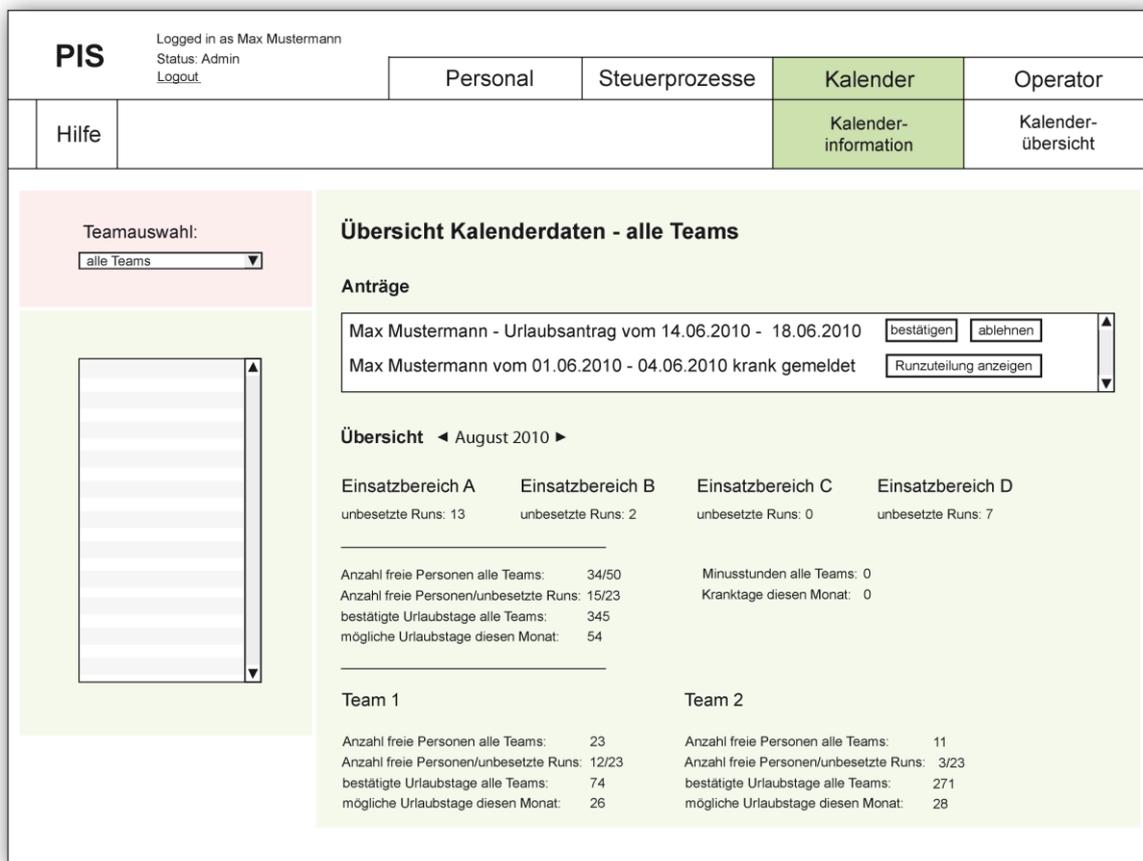


Abbildung 19: Entwurfsmuster der Kalenderinformation eines Teamleiters

Die Kalenderinformation des Mitarbeiters

Der Kalenderinformationsbereich eines Mitarbeiters wird in zwei Hauptaufgaben unterteilt: die Darstellung der zum aktuellen Zeitpunkt laufenden Anträge und das zur Verfügung stellen einer Verwaltungskomponente für Anträge der Mitarbeiter untereinander.

Das Darstellen laufender Anträge erfolgt in chronologisch aufgelisteter Form. Die einzelnen Termine sind jeweils mit einem aktuellen Status (bestätigt, abgelehnt, wird bearbeitet) gekennzeichnet. Alle zum aktuellen Datum zurückliegenden Termine werden fest in der Kalenderübersicht des Mitarbeiters verankert und können nur durch den Teamleiter verändert werden.

Zur selbständigen Verwaltung von Anträgen der Mitarbeiter untereinander wird eine weitere Softwarekomponente angelegt. Hat ein Angestellter zu einem bestimmten Termin keine Zeit, kann er den Dienst mit einem anderen Mitarbeiter tauschen. Dieser Prozess entlastet den Teamleiter bei der Erstellung von Dienstplänen. Der Befragte kann Anfragen auf die Über-

nahme eines Termins beantworten. Ähnlich der Teamleiterverwaltung kann er Anträgen zustimmen oder diese ablehnen. Bestätigt er einen vorliegenden Request wird dieser an den Teamleiter weitergeleitet. Stimmt dieser dem Antrag zu, wird der Kalendereintrag bei dem befragten Mitarbeiter ausgeführt und aus der Kalenderübersicht des anfänglichen Antragstellers gelöscht.

4.6 Konzeption des Operatorbereiches

Um die Steuerprozesse und eine daraus folgende Auswertung der Personalkompetenzen durchführen zu können, müssen die einzelnen Arbeitsplatzressourcen mit für die Bewertung notwendigen Informationen ausgestattet sein. Das Erstellen, Bearbeiten und Zuteilen dieser Hintergrunddaten erfolgt im Operatorbereich und wird durch einen sogenannten „Operator“ ausgeführt.

4.6.1 Aufgaben eines Operators

Der Operator wird als eigene Personengruppe angelegt und erhält mit erfolgreicher Anmeldung Zugriff auf den Operatorbereich. Er hat die Aufgaben Hintergrundinformationen zu erstellen und diese den Arbeitsplatzressourcen zuzuordnen. Seine Arbeit ist dabei grundlegend für das gesamte Personalinformationssystem. Ohne hinterlegte Hintergrundinformationen können keine Steuerprozesse erstellt werden und eine Auswertung der Personalkompetenzen ist nicht möglich. Zur Unterstützung seiner Aufgaben bietet das Personalinformationssystem zahlreiche Hilfetools, welche ähnlich zur Personalverwaltung und zu den Steuerprozessen funktionieren.

4.6.2 Hintergrundinformationen und Runs

Als Hintergrundinformation oder Run werden einzelne Informationspakete angelegt. Ein Informationsblock beinhaltet einen eindeutigen Bezeichner, eine kompakte Beschreibung der auszuführenden Tätigkeit und für die Berechnung der Personalkompetenzen notwendige Parameter. Die Einzelinformationen eines Runs werden aus den in den Personalkompetenzen vorhandenen Informationskomponenten und mit einer durch den Operator festgelegten Wertigkeit zusammengestellt.

Die Hintergrundinformationen lassen sich auf verschiedene Arten in den Steuerprozess integrieren. So können Informationen dauerhaft, für einen bestimmten Zeitraum oder für einen eigens vordefinierten Zeitabschnitt an eine Arbeitsplatzressource gebunden werden. Informationspakete mit einer eigenen Zeitkomponente werden als Run bezeichnet.

4.6.3 Erstellung und Bearbeitung der Hintergrundinformationen

The screenshot shows the PIS interface for managing runs. The user is logged in as Max Mustermann with Admin status. The navigation menu includes Personal, Steuerprozesse, Kalender, and Operator (selected). The 'Runverwaltung' sub-menu is active. The main area displays 'Runinformationen - Run A12345'. On the left, there is a dropdown for 'Einsatzbereich wählen:' set to 'Einsatzbereich A' and a list of runs from A12345 to A12350. The right side shows five competency areas (A-E) with checkboxes and skill levels. For example, in Kompetenzbereich A, Kompetenz 1 and 2 are checked with skill levels of 2 and 5 respectively. At the bottom, there are buttons for 'neuer Run', 'Run speichern', 'Run zurücksetzen', and 'Run löschen'.

Abbildung 20: Entwurfsmuster der Runverwaltung eines Teamleiters bzw. Admins

Das Erstellen, Bearbeiten und Entfernen von Hintergrundinformationen erfolgt in einem in der Beispielanwendung als „Runverwaltung“ bezeichneten Untermenü. Der Operator legt ein neues Informationspaket an und vergibt diesem einen eindeutigen Bezeichner. Eine Mehrfachverwendung der Informationsbezeichnung ist aus programmieretechnischen Gründen nicht möglich. Bei einem falschen Eintrag wird die Runerstellung blockiert und entsprechende Warnhinweise angezeigt. Das Anlegen eines neuen Runs erfolgt nach einem vorgeschriebenen Schema. Die auszufüllenden Eigenschaften werden als eingabepflichtig oder als Alternativeinträge gekennzeichnet. Der Erstellungsprozess kann ähnlich den Personalprofilen durch Runprofile unterstützt werden.

Die Darstellung aller angelegten Runs erfolgt in einer übersichtlichen Liste, in welcher sie zur weiteren Bearbeitung ausgewählt vorliegen. Zur Änderung der Runinformationen stehen zwei Verwaltungsbereiche zur Verfügung: die Runinformationen und die Runkompetenzen.

Runinformationen

Der Bereich der Runinformationen verwaltet alle einem einzelnen Run zugehörigen Daten zur Unterstützung der Steuerprozesse. Neben standardisierten Eigenschaften, wie Bezeichnung und Länge besteht hier die Möglichkeit programmspezifische Angaben zu setzen. Das zuteilen eines Informationsblocks zu einem Einsatzbereich oder einer frei gewählten Kategorie, ist beispielsweise für die farbliche Darstellung der Arbeitsplatzressourcen nutzbar.

Runkompetenzen

Ein zweites Untermenü der Runverwaltung ermöglicht dem Operator das Bearbeiten der Bewertungsparameter einer Hintergrundinformation. Zu diesem Zweck werden ihm alle zur Verfügung stehen Personalkompetenzen angezeigt. Diese können nach Wunsch aktiviert und mit einem zusätzlichen Bewertungsfaktor in der Datenbank speichern werden. Da der Operator nur Komponenten auswählen kann, welche im Personalbereich vorhanden sind, ist eine fehlerhafte Berechnung der Personalkompetenzen ausgeschlossen.

4.6.4 Die Runbesetzung

Ist einem Arbeitsplatz kein Run zugeordnet, findet dieser auch keine Verwendung beim Anlegen von Dienstplänen. Der Operator hat die Aufgabe jedem Arbeitsplatz die zugehörigen Informationen für einen bestimmten Zeitraum zu hinterlegen. Das Personalinformationssystem unterstützt ihn in dieser Funktion.

Das Binden von Informationen an einen Arbeitsplatz kann durch verschiedene Zuordnungsmethoden erfolgen. Hintergrundinformationen können dauerhaft oder für einen bestimmten Zeitraum an eine Arbeitsplatzressource gebunden werden. Bei der dauerhaften Zuteilung verteilen sich die Hintergrundinformationen einmalig auf die Arbeitsplätze und sind ständig für die Steuerprozesse im Einsatz. Der Operator hat bei dieser Methode den geringsten Arbeitsaufwand. Die notwendigen Änderungen erfordern nur das Neuinitialisieren einzelner Arbeitsplatzressourcen. Das temporäre Anlegen von Information gestaltet sich aufwendiger. Zum einen kann der Operator Hintergrundinformationen für einen bestimmten Zeitraum an-

legen und muss nach Ablauf dieser eine Neubelegung vornehmen. Zum anderen kann er sogenannte Runs einsetzen. Diese haben einen bereits vordefinierten Zeitrahmen. Die Arbeitsplatzressource wird nach Ablauf eines Runs automatisch freigegeben.

Die Zuordnung der Hintergrundinformationen erfolgt manuell. Der Operator wählt einen Raum und ein Datum und setzt einen von ihm gewählten Run auf die entsprechende Arbeitsplatzressource. Dieser Prozess ist sehr aufwendig und kann durch das Personalinformationssystem vereinfacht werden. Als Automatisierungsfunktionen bieten sich das Laden eines gespeicherten Belegungsplanes oder ein auf die Runverteilung projizierter Verteilungsschlüssel an. Diese Vorgänge sind der Personenverteilung sehr ähnlich. Aus diesem Grund sind viele Automatisierungsprozesse übertrag- und auf die Runverteilung anwendbar. Es ist darauf zu achten, dass einzelne Runs für mehrere Arbeitsplätze einsetzbar sind. Dies ist beim Verteilen von Personen nicht möglich.

4.7 Fazit der Konzeption

Die hier angelegte Beispielanwendung gibt einen kompakten Einblick zur Konzeption eines Personalinformationssystem. Weitere Bereiche, wie das Anlegen von Personal- bzw. Runstatistiken oder ein softwarespezifischer CMS-Bereich finden in der Konzeption keine Berücksichtigung, können den Funktionsumfang der Anwendung aber zusätzlich erweitern.

5 Realisierung eines Personalinformationssystems

Bei der Realisierung eines PIS werden die durch den Endkunden vorgegeben Kriterien und das daraus resultierende Konzept für den letztendlichen Einsatz der Software in die Praxis umgesetzt. Das folgende Kapitel wird diesen Prozess, die wichtigsten programmiertechnischen Probleme und deren Lösungen dokumentieren. Angelehnt an die Gliederung des vorhergehenden Kapitels werden die wichtigsten Programmteile erfasst, Optimierungsmethoden dargestellt und grundsätzliche Entwicklungsprozesse analysiert. Die programmiertechnische Umsetzung der Beispielanwendung steht dabei nicht im Vordergrund. Vielmehr sollen bei der Entwicklung eines Personalinformationssystems entstehende Standardprobleme in allgemeiner Form diskutiert und unter dem Kriterium möglichst effizienter Lade- und Steuerprozesse erforscht werden.

5.1 Die Datenbank- und Ordnerstruktur

Für ein PIS wird prinzipiell der Einsatz einer Thin-Client-Architektur empfohlen, bei welcher alle notwendigen Daten und Informationen der Anwendung auf einem Server gespeichert sind. Da es schwer ist die einzelnen Softwarekomponenten in ungeordneter Form zu verwalten, gilt es eine geeignete Ordnerstruktur auf dem Server und für das Speichern von Informationen gut strukturierte Tabellensätze in der Datenbank anzulegen. Die einzelnen Module sollten übersichtlich, leicht zu finden und für andere Entwickler nachvollziehbar strukturiert sein. Werden zusätzliche Komponenten ergänzt, sollte diese keinen Einfluss auf bereits bestehende Objekte haben.

5.1.1 Datenbankstruktur der Beispielanwendung

Das Strukturieren der Datenbank ist ein wichtiger Faktor, um eine dauerhafte Übersicht der verwendbaren Informationen beizubehalten. Die Anzahl dieser Informationen kann im Laufe der Zeit eine unüberschaubare Größe erreichen, was wiederum zu Redundanz, dem Speichern nicht verwendeter Daten und damit zur Verschwendung von Speicherplatz führt.

Für eine übersichtliche Darstellung der Datenbank sollten grundsätzlich die zu verwendenden Tabellensätze den einzelnen Inhalts- bzw. den wichtigsten Funktionsbereichen zugeteilt werden. Da eine Gruppierungsfunktion von Tabellen in MySQL nicht existiert, bleibt nur die Möglichkeit die Datenbankinhalte durch eine geschickte Namensvergabe einzelnen Berei-

chen zuzuordnen. Durch einen entsprechenden Bezeichner am Anfang des Namens einer Tabelle, wird diese einer jeweiligen Unterkategorie zugewiesen. In der Beispielanwendung sind das exemplarisch die Gruppe der anwendungsunterstützenden Systemfunktionen (z.B. der Login-Bereich – „_login“) oder die Teilbereiche der jeweiligen Inhaltskomponenten (Personal – „ps_“, Steuerprozesse – „sp_“, Kalender – „cal_“ und Operator – „op_“)⁶⁹.

5.1.2 Die Ordnerstruktur

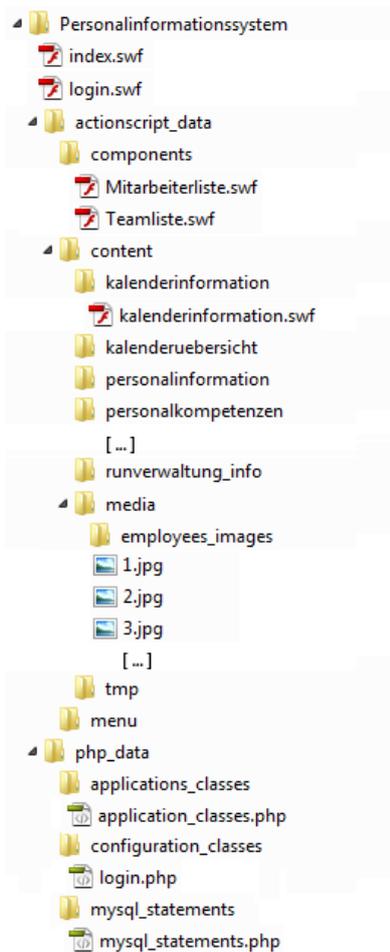


Abbildung 21: Beispielvariante der Ordnerstruktur eines PIS

Das Anlegen der verwendeten Softwarekomponenten in einer geeigneten Ordnerstruktur hat gerade bei der Zusammenarbeit mehrerer Entwicklergruppen eine große Bedeutung. Gesuchte Softwarekomponenten können leichter gefunden und effektiver in die eigenen Programmkomponenten integriert werden. Desweiteren wird die Anzahl ungewollten Überschreibens namensgleicher Dateien auf Grund der besseren Übersicht enorm reduziert.

Für die Gliederung eines Personalinformationssystems sollten mehrere Ordner bestehen, welche die verwendeten Anwendungsschichten repräsentieren. Diese können wiederum nach Funktionalität und Inhaltsbereich in weitere Unterverzeichnisse unterteilt sein.

Die angedachte Beispielanwendung besteht aus zwei Hauptordnern. Der Ordner „actionscrip_data“ enthält alle Daten zur Umsetzung der grafischen Benutzeroberfläche. Der Ordner „php_data“ umfasst Verarbeitungsroutinen, systemrelevante Programmdateien und Softwarekomponenten für die Datenbankabfrage.

⁶⁹ Die detaillierte Darstellung der einzelnen Tabellensätze erfolgt in dem folgenden Kapitel des jeweiligen Anwendungsbereichs.

Der „`actionscript_data`“-Ordner

Zum Anlegen der Benutzeroberfläche werden drei Kategorien benötigt: Ein Verzeichnis zum Anlegen von Anzeigekomponenten (`components`⁷⁰), ein inhaltsabhängiger Programmordner (`content`), in dem unter anderem die Strukturierung der Anzeigekomponente stattfindet, und ein dritter Bereich für extern zu verwaltende Medien (`media`). Für jeden Inhaltsbereich wird ein eigener Unterordner angelegt. Es ist so mögliche zusätzliche Systemkomponenten leichter zu ergänzen und zu verwalten. Die extern angelegten Medien (Bilder, Texte, Videos) sind grundsätzlich in allen Inhaltsbereichen verfügbar.

Der „`php_data`“-Ordner

Im Verzeichnis „`php_data`“ sind alle durch den Server ausgeführten Softwareroutinen angelegt. Diese unterscheiden sich nach Funktionalität und Einsatzbereich und sollten nach ihren Eigenschaften auf einzelne Unterordner verteilt werden. Unterschieden wird in Berechnungs- und Verarbeitungsroutinen (`application_classes`), programmsteuernde Module (`configuration_classes`) und für die Datenbankabfragen zuständige Klassen (`mysql_statements`).

5.2 Realisierung des Anmeldebereichs

Bevor ein Nutzer auf die Programmfunktionen des Personalinformationssystems zugreifen darf, muss herausgefunden werden, ob und welche Zugriffsrechte dieser hat. Der Anmeldebereich realisiert diesen Authentifizierungsprozess. Der Nutzer sendet einen ihm eindeutig zugeteilten Benutzernamen und ein zugehöriges Passwort zur Validierung an den Server. Das serverseitige php-Skript verarbeitet die Anmeldeinformationen, die Angaben werden gegen eine Datenbank geprüft und eine Bestätigung wird zurückgeben.

Aus technischer Sicht sind folgende Aufgaben zu realisieren: Das Erstellen einer Schnittstelle zwischen Actionscript 3.0 und PHP (zum Senden und Empfangen der Benutzerinformationen), das Herstellen einer serverseitigen Verbindung zur Datenbank und die Realisierung der zugehörigen MySQL-Abfragen.

⁷⁰ Die in beiden Hauptbereichen gewählten Namen der einzelnen Verzeichnisse sind lediglich ein Vorschlag für die Beispielanwendung und können grundsätzlich frei gewählt werden.

5.2.1 Die Schnittstelle Actionscript/PHP (URLRequest, URLLoader, URLVariables)

Das Senden und Empfangen von Daten zwischen Actionscript und PHP stellt einen wesentlichen das ganze Projekt betreffenden Prozess dar. Die clientseitige Anwendung muss eine Verbindung zum Server herstellen, Daten zum Senden generieren und beim Empfangen dekodiert den entsprechen Programvariablen zuordnen. ActionScript 3.0 verwendet die URLRequest, die URLLoader- und URLVariables-Klassen zum Laden und Empfangen von Informationen.

Die URLRequest-Klasse erfasst alle Informationen, die zur Verbindungsherstellung einer auf dem Server liegenden Datei notwendig sind. Diese sind neben der zu verwendenden URL unter anderem die HTTP-Übertragungsmethoden GET oder POST. Da GET Informationen sichtbar im Browserfenster überträgt, diese aber keiner anderen Person zugänglich sein sollen, wird für Personalinformationssysteme prinzipiell der Gebrauch der POST-Methode empfohlen. Es gilt zu beachten, dass eine Verbindung zu externen Daten nur herzustellen ist, wenn diese im gleichen Verzeichnis und dessen Unterordnern liegen. Die URLRequest-Objekte werden an die `load()`-Methoden von URLLoader übergeben. Mit dieser wird die Schnittstellendefinition gestartet.

Die URLLoader Klasse stellt die Verbindung zu einer sich auf dem Server befindenden Datei her. Sie verwendet das erweiterte Ereignisverarbeitungsmodell⁷¹ von ActionScript 3.0, welches das Überwachen von Ereignissen wie `complete`, `httpStatus`, `ioError`, `open`, `progress` und `securityError` ermöglicht. Daten der URLLoader-URL stehen erst dann zur Verfügung, wenn die Übertragung der Informationen vollständig abgeschlossen ist. Fehler und Ereignisse können mit dem URLLoader effizienter als bei früheren Actionscript-Versionen verarbeitet werden. Das Ergebnisprotokoll ermöglicht den Fortschritt des Downloads (geladene Byte und Gesamtanzahl der Byte) zu überwachen, was zum Beispiel bei der grafischen Darstellung des Preloading-Prozesses bzw. beim Ladeprozess von Personenbildern zum Einsatz kommt. Geladenen Zeichenfolgen und Systeminformationen werden durch Actionscript in UTF-8 oder UTF-16 entschlüsselt. Zum Anlegen von Sende- bzw. Empfangsvariablen wird die Klasse URLVariables genutzt.

Objekte der URLVariables-Klasse werden ausschließlich in Verbindung mit der URLLoader-Methode verwendet. Sende- bzw. Empfangsvariablen sind als einzelne Objekte angelegt und

⁷¹ Vgl. [hlpadobeUC2010]

mit einem Bezeichner und dem zugehörigen Inhalt initialisiert. Besondere Aufmerksamkeit gilt dem Und-Zeichen (&). Dieses wird innerhalb des URLVariables-Konstruktors als Trennzeichen benutzt und muss bei der Übergabe durch URL-Kodierung von & in „%26“ umgewandelt werden.

Das folgende Beispiel stellt den Datenaustausch zwischen Actionscript und PHP detailliert dar. Für den Anmeldebereich der Beispielanwendung werden zwei URLVariables-Objekte namens „username“ und „password“ gesendet und eine als „access“ benannte Variable empfangen. Das URLRequest-Objekt wird mit der Beispieldatei „login.php“ initialisiert und auf POST eingestellt.

Actionscript:

```
1 // Initialisierung der Sendevariablen
2 var variables:URLVariables = new URLVariables("username=Mustermann");
3 var variables:URLVariables = new URLVariables("password=12345");
4
5 // Einstellen der Schnittstellenparameter
6 var request:URLRequest = new URLRequest();
7 request.url = "http://localhost/login.php";
8 request.method = URLRequestMethod.POST;
9
10 // Anlegen eines Objekts, das die zu empfangenen Daten enthält
11 request.data = variables;
12
13 // Erstellen eines Schnittstellenobjekts
14 var loader:URLLoader = new URLLoader();
15 // Angeben, dass heruntergeladene Daten als URL-codierte Variablen
16 // empfangen werden
17 loader.dataFormat = URLLoaderDataFormat.VARIABLES;
18 // Anlegen eines Eventlistener für die Statusabfrage
19 loader.addEventListener(Event.COMPLETE, completeHandler);
20
21 try
22 {
23     loader.load(request);
24 }
25 catch (error:Error)
26 {
27     trace("PHP Datei konnte nicht gefunden werden");
28 }
29
30 function completeHandler(event:Event):void
31 {
32     // bei erfolgreicher Übertragung Ausgabe der Zugangsvariable
33     trace(event.target.data.access);
34 }
```

Beispiel 1: Schnittstellenverbindung zwischen Actionscript und PHP

Die login.php empfängt die in Actionscript deklarierten Variablen, benutzt diese, um Datenbankabfragen zu starten, und sendet eine Zugangsvariable „access“ an die Actionscript Datei zurück.

PHP:

```
1 // Empfangen der Actionscript Variablen und als PHP Variablen deklarieren
2 $username = (str) $_POST["username"];
3 $password = (str) $_POST["password"];
4
5 [...] //Initialisierung eines Rückgabewertes → $access = true|false [...]
6
7 // Rückgabe der Zugangsdatei an Actionscript
8 echo "access=".$access);
```

Beispiel 2: Empfangen der Actionscriptvariablen und Zurücksenden einer Zugangsvariable

5.2.2 PDO (PHP Data Objects) als Datenbankschnittstelle

Für die Arbeit mit Datenbanken gilt es seitens PHP einen zugehörigen Server und die zu verarbeitende Datenbank zu authentifizieren, sowie eine Verbindung zu diesen herzustellen. Seit PHP 5.1 kann dafür das Datenbankzugriffsobjekt PDO genutzt werden. Das PHP Data Object oder kurz PDO ermöglicht einen einheitlichen Zugriff von PHP auf unterschiedliche SQL-Datenbanken, wie zum Beispiel MySQL 3.x/4.x, ODBC v3 oder SQLite 3.x. Eine Anwendung muss nicht für jede Datenbank umgeschrieben werden, wodurch der Anpassungsaufwand beim Umstieg auf ein anderes Datenbankmanagementsystem minimiert wird. Als Anweisungssyntax nutzt PDO den Befehlssatz von MySQL 5.x⁷² und gewährleistet so einen optimalen Funktionsumfang zur Erstellung eines Personalinformationssystems. Ein allgemeines Beispiel für den Verbindungsaufbau zu einer PIS zugehörigen Datenbank befindet sich im Anhang⁷³.

5.2.3 Das Verwalten der Benutzerrechte

Bei einem erfolgreichen Authentifizierungsprozess werden die Zugriffsrechte der einzelnen Programmdateien geladen. Diese sind für jede Person in der Datenbank gespeichert und nach

⁷² Siehe [MYSQL2010]

⁷³ Siehe Anhang Beispielcode PDO

der Menüstruktur des Personalinformationssystems angelegt. Zur effizienten Ausführung des Ladeprozesses werden die Daten nach einem bestimmten Schema verwaltet.

Grundsätzlich wird für jede Menüebene eine eigene Tabelle erstellt. Als Primärschlüssel gelten die einzelnen Benutzer-IDs. Hat ein Nutzer keinen Zugriff auf eine Untertabelle, wird dessen ID nicht im Primärschlüssel dieser aufgeführt. Die Spalten der einzelnen Tabellen sind mit dem Namen der jeweiligen Untermenütabelle bezeichnet und mit einem numerischen Zuordnungsparameter gefüllt. Wird dieser Wert mit 0 initialisiert, besteht kein Zugriffsrecht auf den jeweiligen Menüunterpunkt. Alle anderen Parameter dienen als Referenzwerte für Tabellen, welche die Menübeschriftung verwalten. Eine rekursive Methode liest die einzelnen Menünamen durch die Kombination der Tabellenwerte und der Namenstabellen in separate Arrays ein und übergibt diese zur weiteren Verarbeitung an die Actionscript-Navigationskomponente. Der Wert 0 wird als Abbruchkriterium der rekursiven Methode genutzt. Die Abfrage wird verkürzt, da nur Untertabellen abgerufen werden, auf denen vorhandene Zugriffsrechte liegen. Tabellen der untersten Menüebene erhalten eine gesonderte Behandlung. In ihnen werden Spalten mit allen für diesen Bereich zur Verfügung stehenden Objekten eines Menüpunktes angelegt. Die Spalten sind direkt mit dem Dateinamen (ohne Dateiendung) der verwendeten Softwarekomponente benannt und mit einem booleschen-Wert initialisiert. Die mit „true“ gefüllten Spaltennamen werden in einer Feldvariable (Array) hinterlegt. Diese wird mit dem zugehörigen Tabellennamen betitelt und an die Actionscript-Basiskomponente übergeben. Die einzelnen Arraynamen sind durch diese Vorgehensweise mit den eingetragenen Werten der für die Menükomponente angelegten Arrays identisch.

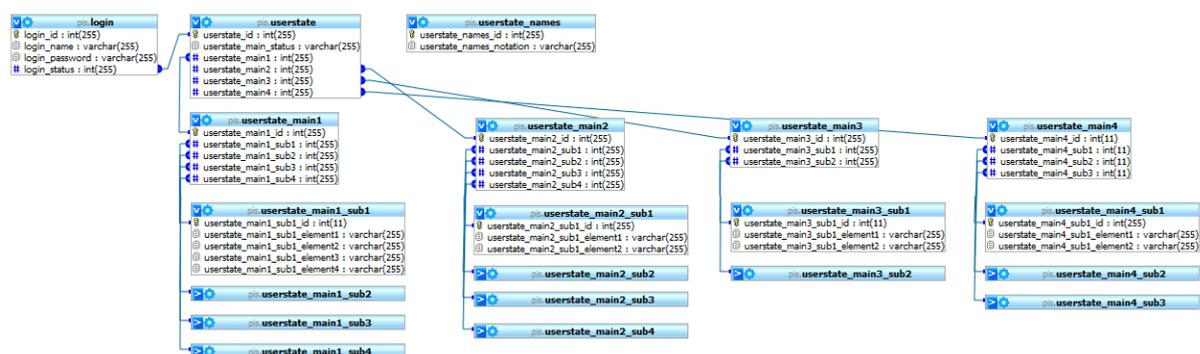


Abbildung 22: Datenbanktabelle des Anmeldebereiches

Übergabe eines Arrays

Das Übergeben von Arrays aus einem PHP-Skript in eine Actionscript-Datei wird durch die URLLoader-Methode nicht unterstützt. Die Arrayinhalte müssen einzeln gesendet und seitens Actionscript zu dem gewünschten Array zusammengefügt werden. Diese Vorgehens-

weise ist sehr unübersichtlich und kann bei Übertragungsindifferenzen zur falschen Übermittlung der im ursprünglichen Array angelegten Reihenfolge führen. Zur Lösung dieses Problems werden die einzelnen Elemente des Arrays serverseitig ausgelesen und in einem String zusammengefasst. Hinter jedem Element wird ein inhaltsunabhängiges Trennzeichen eingefügt, die erstellte Zeichenfolge an Actionscript gesendet und dort mittels der `split()`-Funktion in einen Array zurück gewandelt. Die gewünschte Anordnung wird beibehalten und der Übertragungsprozess durch das Senden einer einzigen Variable zusätzlich verkürzt.

Actionscript:

```
1 private function completeHandler(event:Event):void
2 {
3     var loader:URLLoader = URLLoader(event.target);
4
5     // Empfangen des auf dem Server erstellten Strings -
6     Bsp.: personalinformation="personalinformation|teamauswahl|mitarbeiterauswahl"
7     var variables:URLVariables = new URLVariables(loader.data);
8 }
9
10 // Ergebnis durch das Anwenden von split() -
11 //Bsp.: personalinformation = [personalinformation, teamauswahl, mitarbeiterauswahl]
12 var personalinformation:Array = variables.personalinformation.split("|");
```

Beispiel 3: Empfangen eines String und Weiterverarbeitung zu einem gewünschten Array

5.2.4 Der Direktzugriff auf nichtauthentifizierte Programmdateien

Der Anmeldebereich wird durch den Aufruf einer SWF-basierenden Login-Datei gestartet, er prüft die zugehörigen Nutzerrechte und lädt die zur Arbeit freigegebenen Softwarekomponenten. Nutzer könnten versuchen diesen Prozess zu umgehen, in dem sie Programmdateien direkt über die objekteneigene URL aufrufen. Da Dateien nur mit Inhalt gefüllt werden, wenn sie mit den zugehörigen Eingangsparametern beispielsweise der Benutzer-ID verknüpft sind, lassen sich personengebundene Daten oder andere zu initialisierende Informationen über diese Methode zunächst nicht auslesen. Das Dokument wird leer übergeben und keine Daten freigegeben.

Greift der Nutzer jedoch auf verwaltungstechnische Softwaremodule zu, kann er eigene Benutzerprofile anmelden, diese mit für ihn relevanten Nutzerrechten versehen und durch erneutes Anmelden auf nicht öffentliche Informationen zugreifen. Um auch dies zu verhindern, werden Daten in ihrer Programmstruktur mit einer zusätzlichen Funktion zum Prüfen der Zugriffsrechte erweitert. Diese erkennt die zugehörigen Nutzerrechte und gibt erst dann den weiteren Quellcode der Datei frei. Wird dem Nutzer die Arbeit mit einer Programmkomponente verweigert, wird diese geschlossen und eine Weiterleitung auf die Eingangsdatei initiiert.

In Beispiel 4 ist die Prüfroutine für den Aufruf der Datei „personalinformation.swf“ exemplarisch dargestellt. Die für die Beispielanwendung vom lokalen Rechner geladene Variable „Personal[i]“ wird auf die Existenz und die Zugangsberechtigung des jeweiligen Programmteils geprüft. Eine Zugriffsverweigerung bewirkt das Schließen der aufgerufenen SWF-Komponente mit Hilfe der `removeChild()`-Funktion der `URLLoader`-Klasse und das Aufrufen der Login-Datei. Sind die zugehörigen Nutzerrechte vorhanden, werden die weiteren Programmfragmente der aufgerufenen Datei geladen und ausgeführt.

Actionscript:

```
1 import flash.display.DisplayObject;
2
3 //Aus einem Array des Bereichs Personal werden die Zugangsrechte ausgelesen, die Pos. 3
4 //steht an dieser Stelle beispielhaft für die Softwarekomponente der Personalinformation
5 if (Personal[1] != true)
6 {
7     // entfernen aller Programmdatei
8     _root.removeChild(DisplayObject("personalinformation.swf"));
9     // Laden des Anmeldebereichs
10    _root.addChild(DisplayObject("login.swf"))
11 }
12 else
13 {
14     [...] //Programmfunktionen der personalinformation.swf [...]
15 }
```

Beispiel 4: Prüfroutine auf Zugangsberechtigung der Datei „personalinformation.swf“

5.3 Grundstruktur der grafischen Benutzeroberfläche

In der grafischen Benutzeroberfläche (GUI – graphical user interface) werden alle Methoden zur Darstellung von Informationen, zum Auswählen von Programminhalten und zur Steuerung von Benutzereingaben umgesetzt. Die Anordnung dieser Komponenten basiert auf den Kriterien zwei verschiedener Sichtweisen. Aus Nutzersicht ist eine übersichtliche und verständliche Struktur der Programminhalte zu gewährleisten. Beispiellösungen wurden bereits in dem Kapitel der Konzeption behandelt. Aus der Sicht des Entwicklers sollten sich die verwendeten Softwareinhalte in wenigen Schritten in das Gesamtkonstrukt integrieren und bei Änderungen des Layouts möglichst kompakt versetzen lassen.

Um diese Kriterien zu erfüllen, wird grundsätzlich das Anlegen einer Basisdatei empfohlen. In dieser werden fest definierte Anwendungsbereiche erzeugt und mit den zugehörigen Programmkomponenten initialisiert. In Abbildung 23 ist die Basisdatei beispielhaft mit `index.swf` bezeichnet. Zum Erzeugen fest definierter Bildschirmbereiche eignet sich die von Actions-

cript 3.0 bereitgestellte MovieClip-Klasse. Der Anzeigebereich der Software wird in einzelne Container unterteilt und diese auf unterschiedlichen Ebenen angeordnet. Die wichtigsten Anwendungsbereiche eines PIS sind die Menükomponente zum Auswählen von Programminhalten und ein Content-Bereich zur Darstellung der Programminhalte. Diese lassen sich durch verschiedene für den Betrieb relevante Zusatzbereiche ergänzen.

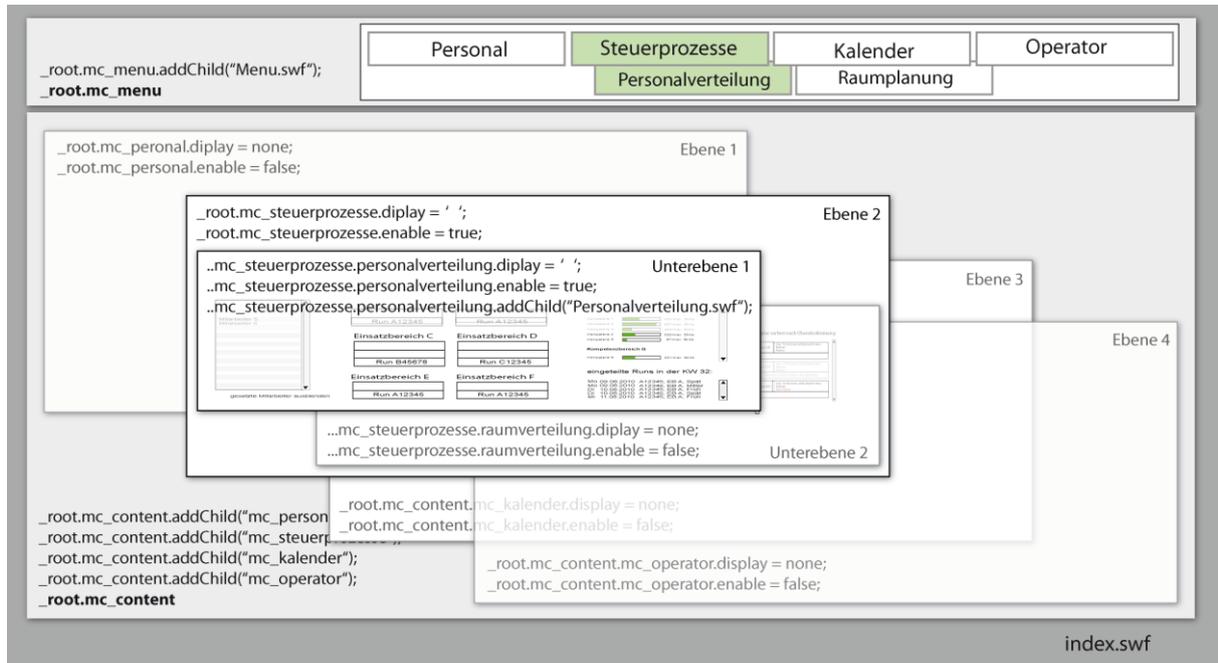


Abbildung 23: Anordnung der MovieClips

5.3.1 Die Movie-Clip Klasse

Zum Anlegen von Bildschirmbereichen benutzt Actionscript das sogenannte MovieClip-Objekt. Ein MovieClip ist eine Abstraktionsebene zum Anlegen von Daten, Dateninhalte und eigenen Containern. Der Ursprung aller Ebeneninstanzen ist das Wurzelement „_root“. Die einzelnen Anzeigeelemente sind durch einen Containernamen und einen Tiefenparameter referenziert und frei auf der Anwenderoberfläche platzierbar. Die dabei verwendeten Koordinaten beziehen sich immer auf die zugehörige Elternebene. Liegen MovieClips übereinander wird der Clip mit dem höheren Tiefenparameter im Vordergrund dargestellt. Anhand dieses Prinzips können mehrere Objekte einem Container zugeordnet und allein durch das Verschieben der Elternebene einheitlich versetzt werden.

5.3.2 Die Menükomponente

Damit der Nutzer gezielt Inhalte der Anwendung auswählen kann, wird eine Menükomponente erstellt. Diese wird in einer eigenen SWF-Datei gespeichert und beim Laden dem „menu“-Container der Basisdatei hinzugefügt. In der Navigationskomponente werden die Kernbereiche der Anwendung auswählbar repräsentiert. Das Untergliedern in weitere Menüunterebenen mit entsprechenden Teilbereichen ist bei Bedarf möglich. Bereits durchgeführte Benutzertests⁷⁴ zeigten, dass eine Menüebene aus höchstens zehn Unterkategorien bestehen und eine maximale Ebenentiefe von sieben Unterschichten aufweisen soll. Beim Überschreiten dieser Wert ist nicht garantiert, dass der Nutzer autonom die Struktur des Programmaufbaus versteht. Die meisten Anwendungen verwenden in der Regel zwei bis drei Navigationsebenen.

Um eine strukturierte Navigationskomponente herzustellen, werden verschiedene Basisinformationen und -komponenten benötigt. Voraussetzungen für das Erstellen eines Navigationsstools sind eigens entwickelte Darstellungskomponenten, die Anzahl der Menüebenen und die Bezeichnung der einzelnen Menüpunkte.

Das Menü ist statisch oder dynamisch erstellbar. Statisch bedeutet, dass der Entwickler den Quellcode für jeden Menüpunkt selbstständig verfasst. Bei der dynamischen Methode initialisiert das Personalinformationssystem die Menükomponente anhand gespeicherter Werte maschinell. Das System erkennt die Anzahl vorhandener Menüunterpunkte und ordnet diese in Verbindung mit einer Darstellungskomponente selbstständig an. Die für diesen Prozess notwendigen Informationen werden in der Datenbank gespeichert, durch ein zugehöriges MySQL-Statement abgerufen und in einzelnen Arrays an den Client übergeben. Um zusätzliche Menüpunkte zu ergänzen, müssen diese lediglich in die Datenbank eingetragen werden. Für jede Menüebene wird ein Array erstellt. In diesem werden die Beschriftungen bzw. die Bezeichnungen der einzelnen Menüpunkte hinterlegt und an den Client geschickt. Die Anordnung der Menüpunkte erfolgt nach der in den Arrays angelegten Reihenfolge. Diese kann direkt aus der jeweiligen Tabelle übernommen und durch entsprechende Sortierfunktionen bei der Datenbankabfrage festgelegt werden.

Zur Umsetzung der Menükomponente wird für jede Ebene eine einzelne Darstellungskomponente angelegt und diese für die zugehörigen Menüpunkte mehrfach genutzt. Sie wird mit einem entsprechenden Bezeichner und ausgewählten Positionierungsparametern initialisiert.

⁷⁴ Vgl. [GruenefLangem1991], 206

Die Wahl des Elternelements bestimmt die anzuzeigenden Untermenüpunkte. Die jeweiligen Darstellungskomponenten sollten grafisch so umgesetzt werden, dass sie bei Mehrfachverwendung ein strukturiertes Gesamtbild der Menükomponente erzeugen.

5.3.3 Die Contentbox

Für die Inhaltsdaten wird ein MoviClip namens „content“ angelegt. In diesem Bereich werden für jede zu integrierende Kategorie eigene Untercontainer erstellt und mit der zugehörigen SWF-Datei initialisiert. Das Anzeigen der Inhalte sollte grundsätzlich auf den gleichen Darstellungsbereich des Monitors projiziert werden, da die Anordnung funktionsähnlicher Komponenten bei sich ständig ändernden Bildschirmpositionen auf den Nutzer verwirrend und unübersichtlich wirkt.

Desweiteren erwartet der Nutzer bei dem Aufruf einer Inhaltskomponente möglichst geringe Wartezeiten. Auf Grund dessen sollten die Inhaltsdateien bei Start des Programms durch einen Preloading-Prozess⁷⁵ lokal gespeichert, in die Basisdatei integriert und vorerst für den Nutzer nicht sichtbar dargestellt sein. Die Auswahl eines Menüpunktes aktiviert die `display()`-Eigenschaften⁷⁶ des zugehörigen MovieClips, deaktiviert alle unerwünschten Inhalte und zeigt den gewünschten Inhalt an. Diese Methode ist bei weitem schneller als das unmittelbare Laden der Inhalte bei Aufruf eines Menüpunktes.

5.3.4 Weitere Programmmodule

Neben der Navigationskomponente und den Dateien der Contentbox werden weitere nicht unbedeutende Softwaremodule zur Ausführung des Personalinformationssystems benötigt. Diese werden ebenso in eigenen Dateien angelegt, bei entsprechenden Nutzerrechten aufgerufen und in die Grundstruktur der Basisdatei oder in die verschiedenen Inhaltsdateien integriert. Um die Entwicklung des Personalinformationssystem möglichst einfach zu halten, müssen die zu entwickelnden Programmmodule gewisse Grundvoraussetzungen erfüllen. Sie sollten mehrfach und vom Anwendungsbereich unabhängig angelegt werden können.

⁷⁵ Vgl. Kapitel 5.4.2 Der Preloading-Prozess

⁷⁶ Vgl. [hlpadobeMC2010]

Die Mitarbeiterauswahl

Die wohl am häufigsten benutzte Zusatzkomponente eines Personalinformationssystems ist die Mitarbeiterauswahl. Für den Teamleiter und den Admin ist sie eine nicht wegzudenkende Komponente. Personengruppen können gezielt aufgerufen, einzelne Mitarbeiter gewählt und die mit ihrer Personen-ID verknüpften Informationen im zugehörigen Inhaltsbereich dargestellt werden.

Die Mitarbeiterkomponente wird inhaltsübergreifend eingesetzt. Das heißt, verschiedene Programmmodule benutzen ein und dieselbe Komponente, wodurch ausgewählte Listeneigenschaften auch bei Inhaltswechsel erhalten bleiben. Die in der Personenliste gesetzten Statusinformationen (z.B. ausgewählter Mitarbeiter) lassen sich lokal (globale Variablen bzw. SharedObjects⁷⁷) oder in der Datenbank speichern und auf die Verarbeitungsroutinen anderer Softwaremodule anwenden.

Die Mitarbeiterauswahl stellt sich meistens in Listenform dar. Sie kann aber auch durch andere grafische Konzepte umgesetzt werden. Die Anordnung der ausgewählten Personen erfolgt nach einer bei der zugehörigen MySQL-Abfrage angegebenen Sortierfunktion. Alle grafischen Objekte der Mitarbeiterauswahl sind in einer eigenen SWF-Datei gespeichert und dort strukturiert. Diese wird beim Laden einem Movieclip zugeordnet und lässt sich anhand dessen Eigenschaften zusätzlich modifizieren. Desweiteren sind einzelne Objekte der Auswahlkomponente mit sogenannten Event-Listenern belegbar. Bei Auswahl eines neuen Mitarbeiters wird beispielsweise ein Personenwechsel (Event.Change⁷⁸) erkannt, die neue Personen-ID eingelesen und Funktionen zum Aktualisieren des Anzeigebereichs ausgelöst.

5.3.5 Das Einbinden externer SWF-Dateien

Die Menükomponente, durch diese auswählbare Inhaltsmodule und einige zusätzliche Programmapplikationen sind in separaten SWF-Dateien gespeichert. Diese können anhand der Benutzerrechte gezielt geladen werden, wodurch der Initialisierungsprozess der Software verkürzt wird. Das Aufrufen von Daten ist in Actionscript 3.0 ähnlich dem Laden von Variablen durch die URLLoader-Klasse gelöst. Es wird eine Zielfile durch den Aufruf einer entsprechenden URL referenziert und einem in der Basisdatei angelegten Container-Objekt

⁷⁷ Vgl. Kapitel 5.3.6 Das Speichern von Komponentenstatements

⁷⁸ Vgl. [adobelivedocEv2010]

zugewiesen. Dieser Prozess verwendet die durch `flash.display` klassifizierte `addChild()`-Methode. Die Eigenschaften der geladenen SWF-Datei übertragen sich automatisch auf den `MovieClip`. Die so entstandenen `MovieClip`-Informationen (z.B. Höhe und Breite) lassen sich zur weiteren Verarbeitung verwenden. Ist der Initialisierungsprozess abgeschlossen, besteht die Möglichkeit die Eigenschaften des Containers zu modifizieren. Die Änderungen übertragen sich direkt auf die geladene Datei. Die in einem `MovieClip` angelegten Elemente sind in wenigen Schritten geschlossen an den Darstellungsbereich anpassbar. Auf diese Weise ist es möglich Daten sehr einfach neu zu positionieren oder im Sinne der Barrierefreiheit⁷⁹ durch entsprechende Skalierungsfunktionen zu verändern.

Ein weiterer Vorteil der `addChild`-Methode ist, dass neben SWF-Dateien externe Bilddateien (z.B. eine JPEG-, GIF- oder PNG-Datei) geladen und nach gleichem Prinzip verarbeitet werden können. Es ist zu beachten, dass im Gegensatz zu einer Bilddatei, SWF-Dateien ActionScript-Code enthalten. Externen SWF-Dateien mit Klassen, die über den gleichen Namensraum verfügen, wie Klassen in der ladenden SWF-Datei, können Namenskonflikte verursachen. Zusätzlich müssen sich extern geladene SWF-Dateien im gleichen Verzeichnis bzw. dessen Unterverzeichnissen der ladenden Datei befinden. Dies ist durch das bereits beschriebene Sandboxverfahren⁸⁰ begründet.

5.3.6 Grundgedanken zur Erstellung der Programmkomponenten

Eine Softwarekomponente besteht in der Regel aus einzelnen grafischen Elementen zum Darstellen von Informationen (Textfelder, Bilder, Vektorgrafiken) und Objekten zum ausführen von Steuerroutinen (Events). Diese Objekte lassen sich mittels der `Graphics`- (zum Anlegen von Vektorgrafiken) bzw. der `Event`-Klasse von Actionscript 3.0 von Grund auf durch den Entwickler eigenständig oder Mithilfe von Inhaltsobjekten der Flash eigenen Komponenten-Bibliothek erzeugen. Beide Techniken besitzen Vor- und Nachteile und sollen in den folgenden Abschnitten diskutiert werden.

Allgemeine Vorteile durch das Nutzen von Komponenten

Durch das Erstellen einer Komponente lässt sich Design und Kodierung voneinander trennen. Der Entwickler generiert grafische Module und besetzt diese mit prozessrelevanten

⁷⁹ Begriffserklärung Barrierefreiheit unter [Schulz2009], 7 ff.

⁸⁰ Vgl. Kapitel 3.3.3 Das SWF-Format und das damit verbundene Sicherheitskonzept

Funktionen. Der Designer arbeitet diese mittels der MovieClip-Technik in die grafische Benutzeroberfläche ein. Durch die Angabe von Parametern kann dieser unter anderem Größe, Position und Verhalten der Komponenten an den zugehörigen Inhalt anpassen. Um nicht alle Eigenschaften einer Komponente neu initialisieren zu müssen, sind grundsätzlich Standardwerte angelegt. Wenn einer Anwendung eine Komponente hinzugefügt wird, benötigt diese einmalig eine bestimmte Größe an Speicherressourcen. Beim Hinzufügen weiterer Komponenten der gleichen Art, nutzen diese denselben zugewiesenen Speicherplatz. Durch diese objektorientierte Vorgangsweise wird eine unnötige Zunahme der Anwendungsgröße vermieden.

Das Verwenden der Actionscript-Komponenten-Bibliothek

Mithilfe von fertigen Actionscript-Komponenten geht die Erstellung robuster Anwendungen mit einheitlichen Gestaltungs- und Verhaltensweisen schnell und einfach. Anstatt benutzerdefinierte Schaltflächen, Kombinationsfelder und Listen manuell zu erstellen, sind bereits definierte Objekte vorhanden. Die verwendeten Komponenten müssen einmalig in einem Bibliotheksbereich der Basisdatei angelegt, können dann durch Actionscript aufgerufen und in ihrer Darstellung bzw. ihrem Verhalten an das Anwendungsdesign angepasst werden. Jede Komponente verfügt über einen eigenen Satz von ActionScript-Methoden, -Eigenschaften und -Ereignissen. Die ActionScript 3.0-Komponenten sind FLA-basierte Dateien (.fla-Dateien) mit einer zugehörigen AS-Datei⁸¹. Diese werden mit der Basisdatei kompiliert und in das ausführbare SWF-Dateiformat automatisch übernommen. Zu den wichtigsten ActionScript 3.0-Komponenten zählen die UI-Komponenten⁸² (Benutzeroberflächenkomponenten) und die FLVPlayback-Komponente (ein Flash eigener Videoplayer).

Der Vorteil beim Verwenden von fertigen Actionscript-Komponenten ist, dass grafische Benutzeroberflächen sehr schnell und mit einfachen Mitteln erstellbar sind. Die Objekte werden einmalig in der Programmbibliothek angelegt und sind dann unter Benutzung des gleichen Speicherbereichs mehrmals aufrufbar. Die aufgerufenen Instanzen erhöhen den Speicherplatz nur geringfügig. Es ist möglich zahlreiche Eigenschaften der Objekte dynamisch zu verändern. Eine Anpassung an die jeweilige Oberfläche lässt sich mit minimalem programmiertechnischem Aufwand durchführen. Zusätzlich sind in die Flash-Komponenten bereits

⁸¹ In dieser einem FLA-File zugehörigen Dateie wird der Actionscript 3.0 Quellcode erfasst.

⁸² Verfügbar sind: Button, CheckBox, ColorPicker, Combobox, DataGrid, Label, List, NumericStepper, RadioButton, ProgressBar, ScrollPanel, Slider, TextArea, TextInput, UILoader und UIScrollbar – die Funktionsweise der einzelnen Komponenten ist unter: **[hlpadobeCo2010]**

zahlreiche Events integriert. Diese sind lediglich mit den gewünschten Funktionen zu verknüpfen, ohne dass eine eigenständige Programmierung von Eventfunktionen gefordert wird.

Für die Entwicklung eines Personalinformationssystem sind vorgefertigte Oberflächenkomponenten nicht geeignet. Die Actionscript-Komponenten sind sehr speicherintensiv und aus diesem Grund für die Verwendung innerhalb einer Client-Server-Architektur eher ungeeignet. Die Ursache für den hohen Speicherbedarf liegt in der Umsetzung der einzelnen Objekte. Diese sind durch Vektorgrafiken auf der Flashbühne gezeichnet, wodurch im Vergleich zur quillcodebasierenden Umsetzung grundsätzlich mehr Speicherplatz benötigt wird. Zusätzlich kann der enorme Funktionsumfang einer Actionscript-Komponente zur weiteren Ressourcenverschwendung führen. Die meisten Funktionen und Events werden von dem Entwickler nicht genutzt, erhöhen aber die Anwendungsgröße. Ein weiterer Nachteil beim Verwenden fertiger Komponenten ist die eingeschränkte Anpassungsfähigkeit der einzelnen Objekte an das Anwendungsdesign. Zwar lassen Größe, Farbe, Schrift und verschiedene Stileigenschaften der Komponente Änderungen zu, die eigentliche Grundgestaltung der Komponente bleibt aber erhalten. Ein eigenes Entwurfsmuster lässt sich nicht auf die vorgefertigten Komponenten übertragen.

Das Erstellen eigener Komponenten

Die Actionscript-API bietet die Möglichkeit, benutzerdefinierte Komponenten völlig neu zu erstellen. Diese werden komplett in Quellcode umgesetzt, in eigenen SWF-Dateien hinterlegt und gezielt anhand der Benutzerrechte in die Anwendungsoberfläche geladen. Es bietet sich an einzelne Komponenten, die permanent den gleichen Nutzerrechten zugeordnet sind, in einer gemeinsamen Datei abzulegen. Da eine zu ladende SWF-Datei einen grundsätzlichen Speichereigenbedarf von 531 Byte⁸³ benötigt, lässt sich so zusätzlicher Speicherplatz einsparen.

Das Entwickeln eigener Softwaremodule erzeugt im Vergleich zu vorgefertigten Actionscript-Komponenten einen höheren Arbeitsaufwand und setzt ausgeprägte Programmierkenntnisse voraus. Der Speicherbedarf der Programmkomponente wird jedoch enorm reduziert. Zum Generieren eigener Komponenten werden die `flash.display`-Klasse und die `flash.events`-Klasse genutzt.

Das ActionScript 3.0-`flash.display`-Paket enthält Klassen visueller Objekte, mit welchen eige-

⁸³ Dieser Wert wurde durch das Anlegen einer komplett leeren SWF-Datei ermittelt.

ne Komponenten angelegt werden können. Die wichtigsten Unterklassen sind unter anderem die Bitmap-Klasse (zum Verarbeiten von Bildern), die bereits erwähnte MovieClip-Klasse und die Graphics-Klasse. Die auch als Zeichnungs-API bekannte Graphics-Klasse enthält einen Satz an Methoden, die Vektorformen erzeugen, und ist die am häufigsten verwendete Klasse beim Erstellen neuer Anzeigeobjekte. Neben einfachen Formen, wie Linien, Kurven, Füllungen und Farbverläufen bietet sie eine breite Palette oft genutzter Methoden zum Erzeugen geläufiger Zeichenobjekte, wie Kreise, Rechtecke und Ellipsen an. Für einen erweiterten Funktionsumfang enthält die Graphics-Klasse zudem Methoden zum Zeichnen quadratischer Bézier-Kurven, welche Verknüpfungen mit den trigonometrischen Funktionen der Math-Klasse ermöglichen. Das Nutzen von geskripteten Vektorgrafiken hat zwei Vorteile. Zum einen wird im Vergleich zu pixelorientierten Grafiken nur ein Bruchteil an Speicherkapazität benötigt und zum anderen gibt es keine Qualitätsverluste bei unterschiedlichen Skalierungsstufen. Das Personalinformationssystem kann mit quellcodebasierenden Vektorgrafiken problemlos an unterschiedliche Bildschirmgrößen angepasst werden, ohne dass die grafische Benutzeroberfläche unscharf oder verzerrt dargestellt wird.

Die Event-Klasse wird als Basisklasse bei der Erstellung von Ereignisobjekten zur Steuerung des Personalinformationssystems verwendet. Jede Komponente sendet Ereignisse aus, wenn ein Benutzer mit ihr in Interaktion tritt. Beim Anklicken einer Schaltfläche wird beispielsweise das Ereignis `MouseEvent.CLICK` erzeugt, ein durch Interaktion mit der Maus ausgelöstes Event. In diesem Fall wird ein durch den Benutzer bedingter Steuerprozess auf Grundlage eines Eingabegerätes (Maus, Tastatur, Touchscreen) verarbeitet. Ebenso lassen sich einzelne Events durch die programmeigenen Komponenten oder durch signifikante Anwendungsprozesse auslösen. Wählt ein Benutzer zum Beispiel einen Eintrag in einer Liste, löst die List-Komponente das Ereignis `Event.CHANGE` aus. Wird ein Ladevorgang einer UI-Loader-Instanz abgeschlossen, würde in diesem Fall ein `Event.COMPLETE`-Ereignis erzeugt. Ereignisse, die mit den einzelnen Event-Unterklassen verknüpft sind, sind in der Flash-Dokumentation⁸⁴ der einzelnen Klassen detailliert beschrieben.

Verarbeitet wird ein Ereignis einer Event-Methode mit Actionscript-Code. Dieser wird abgerufen, wenn das Ereignis eintritt. Um den Quellcode des Personalinformationssystem möglichst übersichtlich zu gestalten, sollten die zu veranlassenden Event-Funktionen einzeln vordefiniert, über eine zugehörige Instanz aufgerufen und in einer eigenen Datei abgelegt werden. Im Sinne der für eine Beispielanwendung vorgeschlagenen Ordnerstruktur erhält diese den Namen „`application_classes.swf`“.

⁸⁴ Vgl. [hlpadobeEv2010]

Das Trennen und Einordnen von Komponentenfunktionen

Um mehreren Entwicklern das Verwenden einzelner Bestandteile einer Softwarekomponente zu ermöglichen, müssen diese in einer allgemeinen Form entwickelt und je nach Verwendungsbereich getrennt von einander in unterschiedlichen Verzeichnissen gespeichert werden. Der Speicherort einer Datei bzw. einer Anwendungsklasse gibt von vornherein eine kompakte Übersicht über die Art und den Funktionsbereich der dort angelegten Daten. So werden die für die Strukturierung einer Komponente verantwortlichen Dateien (die ausführbare SWF-Dateien) grundsätzlich einem der Präsentationsschicht zugehörigen Ordner zugeteilt. Im Sinne der vorgeschlagenen Ordnerstruktur bekommt dieser die Bezeichnung „components“ und wird dem „actionscript_data“-Verzeichnis zugeordnet. In den dort abgelegten Dateien werden die gewünschten grafischen Objekte erzeugt, diese mit entsprechenden Event-Methoden verknüpft und alle notwendigen Funktionen der Einzelkomponente aufgerufen. Die Verarbeitungsroutinen des Personalinformationssystems sollten in einer programm-eigenen Funktionsbibliothek zusammengefasst werden. Diese beinhaltet zum einen Methoden zum Ausführen lokaler Ereignisse und zum anderen Funktionen mit einer hohen Rechenintensivität bzw. Datenbankabfragen. Eine Trennung der Funktionsbibliothek in zwei Teilbereiche ist durchaus sinnvoll. Lokale Funktionsklassen sollten in einer der Präsentationsschicht zugehörigen Datei namens „application_classes.swf“ und die für die Serverroutinen zugehörigen Funktionen in einer der Anwenderschicht zugehörigen Klasse namens „application_classes.php“ gespeichert werden.

Um den Aufruf von serverseitigen Programmabläufen zu realisieren, ist das Erstellen einer zusätzlichen Actionscript-Methode notwendig. Diese erzeugt eine Verbindung zur sich auf dem Server befindenden Funktionsbibliothek und übergibt dieser einen Methoden-Namen sowie funktionsrelevante Eingangsparameter. Mit diesen Informationen werden die gewünschten Programmrouinen (z.B. Datenbankabfrage) auf dem Server aufgerufen und ausgeführt⁸⁵. Die daraus resultierenden Ergebniswerte werden an den Arbeitsplatzrechner zurück gesendet.

Das Speichern von Komponentenstatements

Einige Programminhalte des Personalinformationssystems verwenden verschiedene Module, um den gleichen Informationsgehalt darzustellen. Diese Einzelobjekte können in verschiedene Teilbereiche der Anwendung integriert sein und komponenteneigene Einstellungsmög-

⁸⁵ Vgl. Kapitel 5.5.2 Strukturierung der Datenbankabfragen

lichkeiten anbieten. Um die Arbeitsprozesse des Benutzers zu unterstützen, sollen die in einer Komponente eingestellten Informationen inhaltsübergreifend auf die in einem anderen Teilbereich vorhandenen funktionsgleichen Objekte übertragbar sein. Ein gutes Beispiel ist die Mitarbeiterliste. Während diese im Bereich Personal in detaillierter Listenform eingesetzt werden kann, könnte sie in anderer grafischen Form, zum Beispiel als Bilderliste, in einer eigenen Komponente für den Steuerprozessbereich erstellt werden. Wird der Anwendungsbereich gewechselt, sollte ein selektierter Mitarbeiter in der neu aufgerufenen Komponente ausgewählt bleiben. Derartige Informationen werden im Rahmen dieser Arbeit als Komponentenstatement bezeichnet.

Durch Komponentenstatements lassen sich die Arbeitsabläufe des Personalinformationssystems beschleunigen. Sie sollten jedoch nur temporär für den Ablauf einer Sitzung eingesetzt werden, solange bis der Nutzer abgemeldet wird. Das permanente Speichern der zugehörigen Informationen kann auf Grund des Sicherheitskonzeptes nur über eine Datenbank erfolgen. Jede Ereignisänderung müsste gespeichert und bei einem Inhaltwechsel geladen werden. Die daraus resultierenden Datenbankabfragen würden zu unerwünschten Verzögerungszeiten innerhalb der Arbeitsprozesse führen und sind dem Nutzen von Komponentenstatements nicht angemessen. Beim lokalen Speichern existieren diese Verzögerungszeiten nicht. Auf Grund des Sicherheitskonzeptes sind Daten jedoch nur temporär oder arbeitsplatzabhängig speicherbar. Es existieren zwei technischen Lösungen: das Einsetzen globaler Variablen im Quellcode der Anwendung oder auf dem Arbeitsplatz angelegte, auch als Flash-Cookies bekannte, Local Shared Objects.

Globale Variablen – Der Gültigkeitsbereich einer Variablen ist der Bereich in dem auf eine Variable zugegriffen werden kann. In ActionScript 3.0 sind Variablen meistens dem Gültigkeitsbereich der Funktion oder Klasse zugeordnet, in der sie deklariert sind. Eine globale Variable wird außerhalb einer Funktions- oder Klassendefinition erstellt. Im Sinne der Beispielanwendung wird sie am Anfang des in der Basisdatei deklarierten Quellcodes eingebunden. Globale Variablen sind Komponenten übergreifend einsetzbar, das heißt alle vorhandenen Softwaremodule haben die Möglichkeit den Inhalt einer globalen Variable abzurufen und zu verändern. Beim Beenden der Anwendung, verfallen die durch das Personalinformationssystem veränderten Variablenwerte. Sie stehen nur temporär im Rahmen der aktuellen Sitzung des Benutzers zur Verfügung.

Für Anwendungen mit größerem programmiertechnischem Aufwand, wie beispielsweise das Personalinformationssystem, sind globale Variable nicht zu empfehlen. Sie bieten den schnellstmöglichen Zugriff auf die Informationen, lassen sich aber nicht objektorientiert struk-

turieren bzw. gruppieren. Bei einer zu großen Anzahl an globalen Variablen verliert der Entwickler schnell die Übersicht, wodurch bereits deklarierte oder durch Actionscript vordefinierte Variablen leicht überschrieben werden. Die dadurch verursachten Fehler sind schwer zu finden und nachzuvollziehen. Eine bessere Variante zum Speichern von Komponentenstatements sind sogenannte Local Shared Object.

Local Shared Objects - Die SharedObject-Klasse dient zum Lesen und Speichern begrenzter Datenmengen auf dem Computer eines Benutzers. Lokale gemeinsame Objekte sind mit Browser-Cookies vergleichbar. Sie sind lokal permanent, können aber bei Bedarf auf null gesetzt oder gelöscht werden.

Zum Erstellen eines lokalen gemeinsamen Objekts wird folgende Syntax verwendet:

Actionscript:

```
1 var so:SharedObject = SharedObject.getLocal("selectedEmployee");
2 so.data.selectedEmployee = new Number(5);
3 so.flush();
```

Beispiel 5: Erstellung und Initialisierung eines Lokal Shared Object

In diesem Beispiel wird das gemeinsame Objekt „SO“ erstellt, eine eigene Variable namens „selectedEmployee“ angelegt, diese mit dem Wert 5 initialisiert und ausdrücklich auf der Festplatte gespeichert (flush). Bei einem erneuten Start des Personalinformationssystem sind die Daten weiterhin vorhanden und werden automatisch auf die betreffenden Programmkomponenten projiziert.

Lokale gemeinsame Objekte sind mitunter sehr hilfreich, bringen aber auch einige Beschränkungen mit sich, die beim Entwurf einer Anwendung unbedingt zu berücksichtigen sind. Es kann vorkommen, dass die SWF-Dateien nicht zum Schreiben lokaler gemeinsamer Objekte berechtigt sind oder die in lokalen gemeinsamen Objekten gespeicherten Daten ohne es zu bemerken gelöscht werden. Flash Player-Benutzern bietet sich die Möglichkeit, den Speicherplatz, der einzelnen oder allen Domänen zur Verfügung steht, selbst zu verwalten. Wenn die Benutzer den bereitgestellten Speicherplatz verringern, könnten einige lokale gemeinsame Objekte gelöscht werden. Der Entwickler sollte aus diesem Grund unbedingt Standardwerte innerhalb des verwendeten Objektes deklarieren, womit dieses auch bei einem nicht vorhandenen Local Shared Object initialisiert werden kann.

5.4 Das Initialisieren von Programmdateien und -inhalten

Zum Ausführen des Personalinformationssystems müssen die einzelnen Programmdateien vom Server geladen, strukturiert und mit Inhalt gefüllt werden. Die dadurch entstehende Wartezeit ist für den Nutzer oft unangenehm und auf ein Minimum zu beschränken.

5.4.1 Preloading vs. Livestream

Ein wichtiges Kriterium für das optimale Arbeiten mit dem Personalinformationssystem ist der Zeitpunkt, zu welchem ein Softwaremodul oder der Programminhalt geladen wird. Daten können während des Programmablaufes, hier als „Livestream“-Prozess bezeichnet, oder beim Start des PIS in einem geschlossenen Ladevorgang aufgerufen werden. Ladeprozesse während des Programmablaufs sind möglichst so zu realisieren, dass der Benutzer nicht oder nur mit sehr geringen Verzögerungszeiten konfrontiert wird. Zu lange Unterbrechungen des Arbeitsprozesses werden als störend empfunden und können dazu führen, dass der Nutzer die Software als unbrauchbar betrachtet. Im Gegensatz dazu ist das geschlossene Aufrufen mehrerer Daten in einem sogenannten „Preloading“-Prozess beim Start der Software dem Nutzer durchaus vertraut und zum Laden großer Programmdateien geeignet. Die dabei verursachte Wartezeit sollte jedoch einen gewissen Zeitraum nicht überschreiten.

Die Zuordnung der Daten zu einem dieser Ladeprozesse ist von der Art und dem Umfang der jeweiligen Information abhängig. Daten, die in ihrer Gesamtheit ein sehr großes Speichervolumen beanspruchen aber nur einzeln und temporär aufgerufen werden, sollten dem Livestream-Prozess zugeordnet werden. Ein in Bezug auf das PIS passendes Beispiel sind die Personendaten. Das Laden der gesamten Informationen aller Mitarbeiter ist sehr zeitaufwändig, obwohl jeweils nur sehr kleine den einzelnen Mitarbeiter betreffende Informationspakete im jeweiligen Darstellungsprozess genutzt werden. Daten mit einem sehr großen Speichervolumen, welche aber in ihrer Gesamtheit einzeln bestehen, sollten dem Preloading-Prozess zugeordnet werden. Im Normalfall betrifft dies unter anderem die einzelnen Programmmodule und dauerhaft gesetzte Informationen.

5.4.2 Der Preloading-Prozess

Damit der Nutzer während der Arbeit mit dem PIS mit möglichst wenigen Verzögerungszeiten konfrontiert wird, sollten alle statisch verwendeten Programmdateien beim Start der An-

wendung geladen, strukturiert und mit für die anfängliche Darstellung notwendigen Inhalten initialisiert werden.

Strukturierung der Programminhalte

Um eine gezielte Anordnung der für die Anwendung notwendigen Einzelkomponenten zu erreichen, muss eine feststehende Grundstruktur definiert werden. Zu diesem Zweck wird das Anlegen einer Basisdatei⁸⁶ empfohlen und diese in feststehende Anwendungsbereiche aufgeteilt. Im Falle der Beispielanwendung sind das die Container „content“ und „menu“.

Da die Basisdatei lediglich zum Anlegen der Programmstruktur genutzt wird und keine grafischen Programminhalte und verzögernd wirkenden Datenbankabfragen verwendet werden, erfolgt der zugehörige Ladeprozess vergleichsweise schnell. Aus diesem Grund sollte die Basisdatei grundsätzlich getrennt vom Initialisierungsprozess der Inhaltsdateien aufgerufen und für die animierte Darstellung der zu ladenden speicherintensiveren Softwaremodule genutzt werden. Zu diesem Zweck wird das Anlegen einer weiteren Ebene innerhalb der Basisdatei empfohlen. Diese liegt über allen der Grundstruktur zugehörigen Containern und dient der Darstellung einer den Preloading-Prozess begleitenden Animation in Form einer eigenen Komponente.

Prinzipieller Ablauf des Preloading-Prozesses

Ist die Basisdatei geladen, muss der weitere Ablauf des Preloading-Prozesses nach einer festgelegten Sequenz von Ladevorgängen erfolgen. Nur so können den Initialisierungsprozess begleitende Animationen und das letztendliche Freischalten der Startkomponente fehlerfrei ausgeführt werden. Zum Dokumentieren des genauen Ladestatus (Komponentenname und geladenes Speichervolumen) des aktuell zu ladenden Programmteils wird empfohlen eine Softwarekomponente erst dann aufzurufen, wenn die vorhergehende Datei initialisiert ist.

Die zugehörige Sequenz der zu ladenden Daten definiert sich durch folgende Schrittfolge: das Erzeugen der Preloading-Komponente, das dynamische Aufrufen der Inhaltsdateien und das Laden des Menüobjektes. Auf diese Weise wird garantiert, dass die Preloading-Komponente alle nachfolgenden Softwaremodule exakt dokumentiert und die Menükomponente die in ihr definierte Startkomponente des Inhaltes freischalten kann. Diese kann erst

⁸⁶ Eine mögliche Variante der Anordnung inhaltlicher Komponente wurde in Kapitel 6.3 „Grundstruktur der grafischen Benutzeroberfläche“ bereits detailliert beschrieben.

angezeigt werden, wenn sie fertig geladen ist. Da im Voraus nicht bekannt ist, welches Inhaltsmodul zur Anzeige gebracht werden soll, müssen alle Contentobjekte vor der Menükomponente geladen werden.

Das dynamische Laden der Inhaltsdateien

Damit das Personalinformationssystem flexibel und nach Benutzerrechten gezielt ladend umgesetzt wird, werden die einzelnen Inhaltskomponenten in jeweils eigenen Dateien gespeichert. Durch diese Vorgehensweise ist der Anwendung vor dem Authentifizieren eines Nutzers nicht bekannt, welche und wie viele Inhaltskomponenten aufzurufen sind. Aus diesem Grund muss eine dynamische Funktion zum Laden der Contentobjekte erstellt werden.

Das dynamische Laden der Inhaltsdateien basiert auf in der Datenbank gespeicherten Dateinamen. Diese werden in einem Array übergeben und nacheinander verarbeitet. Um einen geordneten Aufbau der anzulegenden Objekte zu erreichen, sollte ein nah an die Menüstruktur angelehnter Tabellensatz verwendet werden. Dieser wird in kleinere schneller abrufbare Tabellen unterteilt, wodurch die Inhaltsdaten den MovieClips der ihnen zugehörigen Kernbereiche direkt zugeordnet werden können. Werden die Inhaltsdateien gleich mit den Namen der in den Menütabelle verwendeten Bezeichnungen benannt, müssen diese nicht in eigenen Datensätzen gespeichert werden. Die zu verwendeten Arrays liegen so bereits strukturiert vor, wodurch auf eine zusätzliche Datenbankabfrage verzichtet werden kann.

Mit dem vollständigen Übertragen der Dateinamen auf den Arbeitsplatzrechner, kann das Initialisieren der Programminhalte beginnen. Basierend auf den übertragenen Arrays der Navigationskomponente wird für jede Menüebene ein eigener MovieClip angelegt. Die einzelnen Elemente eines Arrays verkörpern den Namen einer jeweiligen Untermenüebene, welche wiederum als Feldvariable vorliegt. Ist diese nicht vorhanden, kann davon ausgegangen werden, dass ein Menüendknoten erreicht ist und das Array-Element den Namen einer Contentdatei symbolisiert. Dieser wird mit der Dateiendung „*.swf“ verknüpft und die zugehörige Datei in den entsprechenden MovieClip geladen.

In Beispiel 6 wird eine Beispielmethode dargestellt, welche den dargestellten Preloadingprozess mit Actionscript 3.0 realisiert:

PHP

```
1 import flash.display.*;
2 import flash.net.URLRequest;
3
```

```

4 // Anlegen eines für den Inhaltbereich zuständigen Containers
5 var content:MovieClip = new MovieClip();
6
7 // Die einzelnen Menü-Arrays werden als Shared Object gespeichert.
8 // Der Name der obersten Menüebene muss bekannt sein.
9 // Das geladene Shared Object liegt bei fertigem Ladenprozess beispielweise in folgender
10 // Form vor:
11 // userstate_main: Array = new Array("Personal","Steuerprozesse", "Kalender", "Operator");
12 var so:SharedObject = SharedObject.getLocal(userstate_main);
13
14 // Aufrufen des Preloading-Prozesses mit dem Array der höchsten Menüebene
15 loadProgramm(SO);
16
17 // Erzeugen des Ladevorgangs
18 function loadProgramm (menuebene) {
19
20     // einzelnes Abrufen und Verarbeiten der Array-Elemente
21     for (var i:int = 0; i < menuebene.length; i++) {
22
23         // Anlegen eines für das aktuelle Array-Element zugehöriges Containerobjekt
24         var menuebene[i] = new MovieClip();
25         // Zuweisen des Container-Objekts zu der zugehörigen Elternebene
26         // "this" bezieht sich immer auf den zuletzt erstellten MovieClip
27         this.addChild(menuebene[i]);
28
29         // Anfragen, ob ein eigenes Array des aktuell ausgewählten Elements existiert
30         var SO = SharedObject.getLocal(menuebene[i]);
31
32         // wenn Unterebene nicht vorhanden
33         if (SO.size == 0) {
34
35             // Laden der zugehörigen Inhaltsdatei
36             var loader:Loader = new Loader();
37             var request:URLRequest = new URLRequest("http://localhost/Personalinformations
38             systems/actionsript_data/content/"+menuebene[i]+"/"+menuebene[i]+".swf");
39             loader.load(request);
40             this.addChild(loader);
41         }
42         else {
43
44             // erneutes Ausführen der Methode um die entsprechende Unterklasse zu
45             //initialisieren
46             loadProgramm(SO);
47         }
48     }

```

Beispiel 6: Realisierung des Preloadingprozesses

5.5 Die Umsetzung des Personalbereichs

Nach der Erstellung eines Anmeldebereichs, der Anwendungsstruktur und des Initialisierungsprozesses, gilt es die einzelnen Inhaltsbereiche zu realisieren. Einer der wichtigsten Kernbereiche des Personalinformationssystems ist der Personalbereich. In diesem werden

fest zugeordnete Mitarbeiterinformationen dargestellt und bearbeitet. Das Konzept der Beispielanwendung beschreibt zu diesem Zweck die vier Teilbereiche Personalinformation, Personalkompetenzen, Verwaltung der Personalinformationen und Verwaltung der Personalkompetenzen.

Diese Teilbereiche werden jeweils in eigenen Dateien gespeichert und durch einen nahezu gleichen Grundaufbau erstellt. Es werden verschiedene Darstellungskomponenten dem Anwender entsprechend angeordnet und anhand einer aufgerufenen Personen-ID mit Inhalt initialisiert. Im Bereich der Personalinformation und der Personalkompetenzen bestehen diese hauptsächlich aus Anzeigeobjekten (Textfelder oder Bilder), während die verwaltungstechnischen Bereiche Editierobjekte benutzen. Die Speicherung der zu initialisierenden Daten erfolgt in einer Datenbank. Die Anwendungsbereiche der Personalinformation (Darstellungs- und Verwaltungsbereich) sowie die Bereiche der Personalkompetenzen benutzen jeweils dieselben Tabellensätze.

5.5.1 Die Datenbanktabellen des Personalbereichs

Zum Speichern der Personalinformationen bzw. -kompetenzen werden für beide Teilbereiche jeweils eigene Tabellensätze angelegt. Diese werden spaltenweise mit den für einen Mitarbeiter möglichen Eigenschaften deklariert und durch die personenzugehörigen Benutzer-IDs zeilenweise differenziert.

Es kann davon ausgegangen werden, dass der Datensatz der Personalinformation bzw. der Personalkompetenzen eine sehr große Anzahl an Informationen enthält. Je mehr Mitarbeiter ein Unternehmen betreut, umso mehr Datensätze müssen verwaltet werden, was wiederum zu länger dauernden Datenbankabfragen führt. Um diesen Prozess entgegen zu wirken, wird empfohlen, den angelegten Personenkreis durch einen möglichst feststehenden redundanten Faktor zu gruppieren und in einzelnen Tabellen zu speichern. Auch redundante Werte, wie unter anderem die Vergütungsgruppe, die Anzahl zu leistender Wochenstunden oder die Teamzugehörigkeit sollten in weiteren Einzeltabellen angelegt und durch einen Referenzwert mit den Mitarbeitertabellen verbunden werden. Gezielte und geschwindigkeitsoptimierende Datenbankabfragen werden so ermöglicht. Ein weiterer wichtiger Faktor beim Anlegen der Personaltabellen, ist das Vordefinieren einzelner Spalten. Durch diese Einstellungen wird das Speichervolumen einer Tabelle reduziert, was wiederum zu optimierten Abfrageroutinen führt. Es sollte immer der kleinstmögliche Datentyp eingestellt werden. So stellt beispielsweise die Postleitzahl immer einen fünfstelligen numerischen Wert dar, welcher durch den

Datentyp MEDIUMINT ohne Vorzeichen beschrieben werden kann. Dieser benötigt im Vergleich zu dem größtenteils benutzten INT-TYP ein Byte weniger Speicherplatz und realisiert einen Wertebereich von 0 bis 16777215.

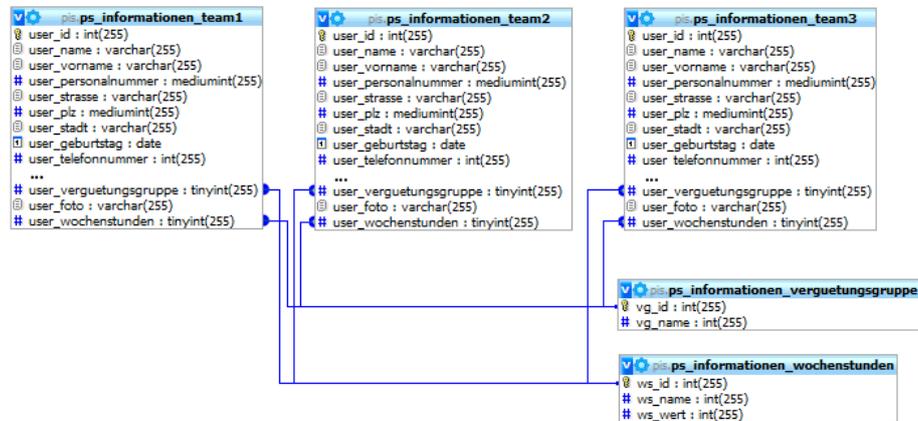


Abbildung 24: Tabellensatz der Personalinformation

5.5.2 Strukturierung der Datenbankabfragen

Um Informationen aus der Datenbank zu erhalten, müssen Datenbankabfragen, welche auch als SQL Queries bezeichnet werden, erstellt werden. SQL Queries sind durch PHP ausführbare Ereignisse, welche das Abrufen bzw. Verändern von Inhalten einer Datenbank realisieren. Die Anwendung der SQL Befehle wird in zahlreichen Dokumenten ausführlich beschrieben und in dieser Arbeit nicht näher erläutert.

Viele Datenbankabfragen sind in ihrer Art identisch und können für die Zusammenarbeit mehrerer Entwicklergruppen in allgemeiner Form erstellt werden. Es sollte erreicht werden, dass bei der Erstellung eines PIS standardisierte Abfrageroutinen abrufbar sind und diese nur durch die Übergabe relevanter Parameter ausgeführt werden. Zu diesen Übergabewerten zählen unter anderem der abzufragende Tabellename, für die Abfrage relevante Spaltennamen und der zugehörige Zeilenwert. Beispiel 7 stellt eine entsprechende Methodenstruktur mehrerer Datenbankabfragen dar. Im Rahmen dieser Arbeit wird die zugehörige Datei mit „mysql_state-ments.php“ bezeichnet und im Ordner „php_data“ hinterlegt.

PHP

```

1 // Übergabe des Funktionsnamen und eines für die abfrage relevanten Parameters
2 $functionName = $_POST['functionName'];
3 $username = $_POST['username'];
4

```

```

5 // switch-case-Abfrage zu Auswahl der passenden Abfragemethode
6 switch($functionName) {
7
8     case "loadTeams":
9         loadTeams();
10        break;
11
12    case "verifyUsername":
13        countPersons();
14        break;
15    [...]
16
17    default:
18        echo "keine Funktion übergeben";
19        break;
20    [...]
21
22 // Methode zum Ausführen einer dynamischen Datenbankabfrage – In diesem Fall wird der
23 //Benutzername übergeben und das zugehörige Benutzer-Passwort an den Client geschickt
24 function verifyUsername($username) {
25
26     $sql = "SELECT login_password FROM personaldaten WHERE login_name = '$username'";
27     $result = mysql_query($sql) OR die(mysql_error());
28     $row = mysql_fetch_assoc($result);
29     echo "&userCount=".$row[' login_password ']."&";
30 }
31 [...]

```

Beispiel 7: Auszug aus mysql_statements.php, in dieser Datei werden Datenbankabfragen realisiert und strukturiert

Für den Zugriff auf eines dieser Statements erstellt der Entwickler lediglich eine URLLoader-funktion. Dieser übergibt er den Methodennamen und die zugehörigen Parameter.

Actionscript

```

1 // Senden des Funktionsnamen und des Benutzerpasswortes
2 var variables:URLVariables = new URLVariables("functionName=verifyUsername");
3 var variables:URLVariables = new URLVariables("username=Mustermann ");

```

Beispiel 8: Actionsript-seitige Übergabe von Parametern für den Aufruf eines MySQL Statements

5.5.3 Der Umgang mit Bilddateien

Der Einsatz von externen Bildern kommt in vielen Bereichen des Personalinformationssystems zur Anwendung. Neben der Gestaltung der grafischen Benutzeroberfläche können die unterschiedlichen Bildformate (zum Beispiel JPEG, GIF, BMP) unter anderem zur Verfeinerung der Personendarstellung benutzt werden. Im Gegensatz zu Textinformationen benötigen Bilder ein viel größeres Speichervolumen, sie besitzen unterschiedliche Kenngrößen

und sind nicht direkt in der Anwendung implementiert. Daraus resultierend beschreiben die folgenden Abschnitten grundsätzliche Varianten beim Verwenden von Bildern.

Preloading oder Direktzugriff

Bilddateien benötigen vergleichsweise viel Speicherplatz, weshalb dazu geraten wird, diese in den Preloading-Prozess einzubauen. Sie liegen dadurch lokal vor und garantieren den schnellstmöglichen Zugriff. Bildformate zur Umsetzung der grafischen Benutzeroberfläche sollten grundsätzlich in diesen Prozess integriert werden. Das Vorladen von Bildern ist nicht für alle Bildladeprozesse geeignet. Beispielsweise beim Benutzen von Bildern im Bereich der Personendarstellung ist diese Methode nicht zu empfehlen. Die große Anzahl an Dateien würde zu einem zu langen Preloading-Prozess und zu unnötigen Ladevorgängen führen. Unnötig deswegen, weil der Aufruf der zu verwendeten Bilder gezielt anhand des gewünschten Mitarbeiters erfolgt und selten der komplette Datenbestand im Rahmen einer Sitzung abgefragt wird. Beim direkten Laden von Image-Dateien während des Arbeitsprozesses erzeugt ein Personalwechsel kurze aber unangenehme Wartezeiten. Diese gilt es für den Benutzer möglichst anschaulich durch ladebegleitende Animationen darzustellen.

Die Bildskalierung oder das Anlegen mehrerer Bildgrößen

Beim Laden von externen Bilddateien kann nicht davon ausgegangen werden, dass diese in einer einheitlichen Größe vorliegen. Um eine für die Darstellung im Personalinformationssystem angemessene Format zu erreichen, sind die Kenngrößen des Bildes einzulesen und an den Inhalt der Benutzeroberfläche anzupassen.

Eine mögliche Variante ist das Anwenden einer durch Flash angebotenen Skalierfunktion. Das Bild wird eingelesen, an einen MovieClip gebunden und dessen Breite und Höhe fest durch den Nutzer angegeben. Die Eigenschaften des MovieClip übertragen sich automatisch auf die Darstellung seines Inhaltes und das Bild liegt in der geforderten Größe vor. In der Praxis hat sich diese Technik der Bildanpassung nicht bewährt. Die Übertragung der MovieClip-Informationen führt zu leichten für den Nutzer sichtbaren Verzögerungszeiten und wird im Einzelfall bei zu schnellem Personenwechsel komplett außer Kraft gesetzt. Die Darstellung der ursprünglichen Größe des Bildes ist die Folge, was zu einer kompletten Destrukturierung der Benutzeroberfläche führen kann.

Ein weitaus stabilerer Ansatz ist das Laden eines Bildes in der jeweils für die Benutzeroberfläche geforderten Größe. Die Image-Datei wird beim Speicherprozess mehrfach kopiert und

durch die von PHP angebotene `$_FILES`-Methode in die passende Form gebracht. Den Einzelbereichen der Anwendung werden die zugehörigen Dateipfade zugeteilt und das entsprechende Bild in der geforderten Größe geladen. Ergänzen sich die neu erstellten Bilder zu einem größeren Speichervolumen als das der Ursprungsdatei, verbraucht diese Methode mehr Speicherplatz auf dem Server. Der zugehörige Ladeprozess einer Bilddatei gestaltet sich jedoch in den meisten Fällen um einiges schneller, da die zu verwendenden Bilder größtenteils in kleinerer Form als in ihrem Originalzustand eingebunden sind und dementsprechend weniger Datenvolumen geladen wird.

Filesharing

Das Uploaden von Daten ist in Flash auf Grund des Sandboxverhaltens grundsätzlich nur für Dateien des gleichen Verzeichnisses bzw. dessen Unterverzeichnissen möglich. Um Dateien aus einem externen Verzeichnis zu laden, sind durch PHP realisierte Methoden notwendig. Es ist darauf zu achten, dass das Ausführen serverseitige Filesharing-Funktionen nur dann möglich ist, wenn die betreffende Datei von keiner clientseitigen Anwendung benutzt wird. Ist ein Bild in den aktuellen Darstellungsprozess des PIS integriert, muss dieses erst aus der Anwendung entfernt und dann durch die neu zu integrierende Datei überschrieben werden. Um ein ungewolltes Überschreiben von Personenbildern zu vermeiden, sind diese in eine den Mitarbeitern eindeutig zugeordnete Bild-ID umzubenennen.

5.5.4 Standardeingabekomponenten der Informationsverwaltung

Im Verwaltungsbereich der Personalinformationen bzw. der Personalkompetenzen wird Benutzeroberfläche mittels einzelner Editierkomponenten erstellt, diese mit Personendaten initialisiert und durch die Bearbeitungsmodule verändert und gespeichert. Auf Grund der Tatsache, dass verschiedene Informationselemente bestimmte Inhaltstypen wiedergeben, existieren verschiedene Bearbeitungskomponenten mit vordefinierten Eingabeeigenschaften. Die Bearbeitung des Inhalts wird so vereinfacht und fehlerhafte Eingaben von vornherein vermieden. Die folgenden Editierkomponenten kommen im Rahmen eines Personalinformationssystems sehr häufig zum Einsatz:

TextInput

Die TextInput -Komponente (Texteingabefeld) ist eine einzeilige Textkomponente, die das Eingeben von Informationen ermöglicht. Sie kommt in fast allen Bereich der Informationsbearbeitung zum Einsatz und kann zur Vermeidung fehlerhafter Eingaben durch die Anzahl und die Art der verwendeten Zeichen vordefiniert werden.



List und Combobox

Die List -Komponente ist ein Listenfeld mit Bildlaufleiste zur Auswahl eines oder mehrerer Elemente. In Listen können Textelemente, Grafiken und andere Komponenten dargestellt werden. Eine ComboBox -Komponente (Kombinationsfeld) ermöglicht dem Anwender die Auswahl einer einzelnen Option aus einer Dropdownliste.

Beide Komponenten kommen grundsätzlich dann zum Einsatz, wenn ausnahmslos vorgegebene Daten durch den Benutzer in ein Informationsfeld eingetragen werden. In der Personalinformation betrifft das beispielsweise die Teambestimmung.



Kalenderbox

Das Eingeben von Kalenderinformationen (Geburtsdatum, Firmeneintrittsdatum, usw.) erfordert ein grundlegendes Eingabeformat. Wird dieses nicht eingehalten, kommt es zur Störung verschiedener Verarbeitungsprozesse. Die Fehlerquote ist bei der freien Eingabe von Datumsinformationen sehr hoch. Das Datumobjekt ermöglicht dem Nutzer mittels einer Kalenderübersicht das gesuchte Datum auszuwählen und die notwendige Darstellungsform automatisch zu übertragen.



NumericStepper

Die NumericStepper -Komponente (Zähler) ermöglicht es dem Benutzer, eine geordnete Zahlenmenge schrittweise durchzugehen. Ein vordefinierter Wertebereich kann nicht überschritten werden. Es wird somit garantiert, dass die für die Anwendung geforderten Datentypen und deren zugehöriger Wertebereich initialisiert werden. Eingesetzt werden diese Komponenten hauptsächlich im Bereich der Personalkompetenzen zur Bewertung eines Mitarbeiters.

Abbildung 25: Häufig genutzte Editierkomponenten in einem Personalinformationssystem

Eingabewerte begrenzen

Die Datenbankoptimierung des Personalbereichs beinhaltet das Speichern von Werten im möglichst kleinsten Datentyp. Für das ordnungsgemäße Erfassen der zu übergebenden Werte und ein Nichtüberschreiten des für einen Speichertyp vorgeschriebenen Wertebereichs, muss die Anzahl der einzugebenden Zeichen eines Eingabefeldes auf die passende Größe beschränkt werden. Actionscript bietet zu diesem Zweck verschiedene Komponenten-Methoden an. Eine oft genutzte Funktion ist unter anderem die `maxChars`-Methode, welche die maximale Anzahl eingebbarer Zeichen bestimmt. Das folgende Beispiel verdeutlicht dies anhand der Postleitzahlen. Diese werden auf sechs Zeichen beschränkt und dadurch mit einer an den `MEDIUMINT`-Typ begrenzten Größe übergeben.

Actionscript

```
1 PLZ_TextInput.maxChars = 6;
```

Beispiel 9: Begrenzung des Eingabefeldes der PLZ auf sechs Zeichen

Die restrict-Methode zum Verhindern unzulässiger Einträge

Nicht nur bei der Zeichenanzahl sondern auch bei der Wahl eines zu deklarierenden Zeichentyps kann der Nutzer fehlerhafte Eingaben erzeugen. Werden beispielsweise numerische Werte verlangt, aber eine String basierende Zeichenfolge übergeben, kommt es häufig zu fehlerhaften Berechnungsfunktionen bzw. Speicherroutinen. Aus diesem Grund ist das Begrenzen der Eingabefelder auf die für ihre Weiterverarbeitung zulässigen Zeichen notwendig. Actionscript realisiert dies mit der `restrict`-Methode. Durch ein von Flash vorgegebenes Schema kann anhand dieser Funktion mit einfachen Mitteln eine Einschränkung der verwendbaren Zeichen eingestellt werden. Die folgende Abbildung zeigt die wichtigsten Quellcodebeispiele der `restrict`-Methode.

Actionscript

Beispiel 10: Begrenzung der Eingabe auf Großbuchstaben, Leerzeichen und Zahlen.

```
Strasse_TextInput.restrict = "A-Z 0-9";
```

Beispiel 11: Es werden alle Zeichen bis auf Kleinbuchstaben akzeptiert.

```
Abteilung_TextInput.restrict = "^a-z";
```

Beispiel 12: "^" kann an beliebiger Stelle im String verwendet werden, um zwischen dem Ein- und Ausschluss von Zeichen hin und her zu wechseln. Im folgenden Beispiel werden alle Großbuchstaben außer Q akzeptiert.

```
Name_TextInput.restrict = "A-Z^Q";
```

Beispiel 13: Es kann die Escape-Sequenz u zum Erstellen von restrict-Strings verwendet werden. Im folgenden Beispiel werden nur die Zeichen zwischen ASCII 32 (Leerzeichen) und ASCII 126 (Tilde) akzeptiert:

```
Passwort_TextInput.restrict = "\u0020-\u007E";
```

5.6 Die Implementierung von Steuerprozessen

Der Kernbereich Steuerprozesse dient dem Anlegen und Koordinieren betriebsbedingter Arbeitsabläufe. Im Konzept der Beispielanwendung werden zwei der wichtigsten Steuerprozesse angesprochen: Die Personalverteilung, um Personen den vorhandenen Arbeitsplatzressourcen zuordnen zu können, und die Raumplanung, zum Koordinieren vorhandener Arbeitsbereiche. Das Personalinformationssystem hat die Aufgabe den manuellen Verteilungsprozess zu unterstützen und diesen bei Bedarf durch automatisierte Funktionen komplett zu ersetzen.

Betriebsbedingte Steuerprozesse basieren prinzipiell auf den Eigenschaften vorhandener Arbeitsplätze und den Mitarbeitern zugeteilten Kompetenzen bzw. Anforderungen. Anhand dieser Informationen wird das Personal auf die Arbeitsplätze verteilt und der tägliche Produktionsprozess eines Unternehmens durchgeführt. Die Anordnung der Mitarbeiter ist zeitabhängig und bewirkt durch Länge, Intensivität und Art der vollzogenen Tätigkeit eine mögliche Neubewertung des eingesetzten Personals.

5.6.1 Die Tabellenstruktur der Steuerprozesse

Zum Ausführen der Aufgaben der Steuerprozesse werden prinzipiell zwei kalenderabhängige Tabellen benötigt. Eine erfasst zeitabhängig die zugeteilten Mitarbeiter eine andere die einem Arbeitsplatz zugehörigen Informationen. Der Aufbau beider Tabellen ist dabei identisch. Die Spalten werden durch die Arbeitsplatzbezeichnungen deklariert, mit einem durch den Arbeitsablauf eines Unternehmens definierten Datumsschlüssel zeilenweise referenziert und mit den Werten der Benutzer-ID (Mitarbeitertabelle) bzw. der Run-ID (Tabelle der Arbeitsplatzinformationen) gefüllt.

Eine mögliche Initialisierung des Datumsschlüssels kann beispielsweise über die Kalenderwerte Jahr, Monat, Tag erfolgen. Finden in einem Betrieb tagesbedingte Unterteilungen statt, werden diese zusätzlich zu dem Datumsschlüssel ergänzt. Dies wird unter anderem bei Schichtensystemen verlangt. In Abbildung 26 wird eine Beispieltabelle der Personenverteilung dargestellt. Der Referenzschlüssel berücksichtigt hier das Vorhandensein einer Früh- (1), einer Mittel- (2) und einer Spätschicht (3).

sp_personal_date	sp_personal_pl1	sp_personal_pl2	sp_personal_pl3	sp_personal_pl4	sp_personal_pl5
201005101	2	3	8	5	15
201005102	1	9	6	13	12
201005103	7	14	4	11	10
201005111	9	1	8	2	15
Datumsschlüssel für den 11. Mai 2010 Frühschicht ...					
201005201	5	1	8	2	15

Abbildung 26: Tabellenauszug einer möglichen Dienstplanverteilung der Mitarbeiter

Um zu große Tabelle zu vermeiden, sollten diese in weitere Untertabellen aufgeteilt werden. Unterteilungskriterien sind unter anderem in einem Unternehmen bestehende Arbeitsbereiche bzw. den Mitarbeitern zugeordnete Abteilungen.

5.6.2 Manuelles Zuteilen von Personen

Damit der Arbeitsprozess mit dem Personalinformationssystem so einfach wie möglich gehalten wird, gilt es die manuelle Zuteilung der Mitarbeiter auf die Arbeitsplätze effektiv und mit wenigen Handgriffen zu realisieren. Es existieren für diesen Prozess zahlreiche Umsetzungsvarianten. Die Zuteilungsmethoden der Mitarbeiterverteilung durch direktes Auswählen der Arbeitsplätze und die „Drag and Drop“-Funktion sind die wohl am häufigsten benutzten Realisierungsarten.

Das Aufrufen eines Arbeitsplatzes mit direktem Eintragen eines Mitarbeiters

Eine mögliche Variante der manuellen Mitarbeiterverteilung ist das Auswählen eines Arbeitsplatzes und das direkte Eintragen eines Mitarbeiters. Zu diesem Zweck wird ein Arbeitsplatz angeklickt und ein ihm zugehöriges Editierfenster geöffnet. In dieses wird der Mitarbeiter per Hand eingetragen.

Diese Zuordnungsvariante ist sehr fehleranfällig, da falsche Eingaben grundsätzlich nicht von der Anwendung erkannt werden. Außerdem erkennt das System nicht, ob der Mitarbeiter bereits auf einem anderen Platz gesetzt ist. Um einen sicheren Programmablauf zu garantieren, sind zusätzliche Prüfroutinen erforderlich. Eine einfache Lösung ist das Implementieren einer Mitarbeiterliste in die Editierkomponente. Verhinderte Personen können von der Anzeige dieser ausgeschlossen werden. Desweiteren wird das korrekte Eintragen auf Grund der Direktauswahl einer Person garantiert. Diese Art der Zuteilung gestaltet sich jedoch sehr unübersichtlich, da die Gesamtdarstellung aller Arbeitsplätze teilweise verdeckt wird.

„Drag and Drop“-Eingaben

Eine geeignetere Variante ist das Auswählen eines Mitarbeiters aus einer permanent angelegten Mitarbeiterliste. Diese wird bei Änderungen des Dienstplanes automatisch aktualisiert, das heißt zugeteilte Personen werden aus der Liste entfernt. Die Mitarbeiter werden einzeln ausgewählt und bei gedrückter Maustaste auf den gewünschten Arbeitsplatz gezogen.

Bei der Drag & Drop-Interaktion klickt ein Benutzer auf ein Objekt und verschiebt es mit dem Mauszeiger, bis die Maustaste wieder losgelassen wird. Es werden zwei Mausereignisse verwendet: das Drücken der Maustaste und das Loslassen dieser. Beim Drücken der Maustaste wird das Objekt angewiesen, dem Mauszeiger zu folgen. Wird diese losgelassen wird

ein drittes Mausevent ausgelöst. Die sogenannte `hitTest()`-Methode⁸⁷ erkennt über welchem Objekt sich die aktuelle Mausposition befindet und ordnet dieses dem ausgewählten Mitarbeiter zu. Die Mausereignisse werden durch einen eigene Flash-API⁸⁸ zur Verfügung gestellt, und könnten durch folgenden Beispielquellcode umgesetzt werden.

Actionscript:

```
1 import flash.events.MouseEvent;
2
3 // Diese Funktion wird aufgerufen, wenn die Maustaste gedrückt wird.
4 function startDragging(event:MouseEvent):void {
5     Mitarbeiter_Listenbild.startDrag();
6 }
7
8 // Diese Funktion wird aufgerufen, wenn die Maustaste losgelassen wird.
9 function stopDragging(event:MouseEvent):void {
10    Mitarbeiter_Listenbild.stopDrag();
11 }
12
13 Beispiel 14: Drag and Drop Ereignisse beim Zuteilen eines Mitarbeiters
```

Befindet sich eine Drag-Komponente beim Loslassen über mehreren zuordenbaren Arbeitsplätzen, muss erkannt werden, welcher davon ausgewählt werden soll. Es müssen zusätzliche Annäherungsfunktionen bereitgestellt werden. Beispiel 15 zeigt exemplarisch die Berechnung der Entfernung zwischen den Objektkoordinaten und der aktuellen Mausposition.

Actionscript:

```
1 // ein Koordinatenobjekt wird angelegt
2 var pointMC1:Object = {x:0, y:0};
3
4 // Koordinatenobjekt wird durch Arbeitsplatzkoordinaten initialisiert
5 get(Parent+".Arbeitsplatz1_mc").localToGlobal(pointMC1);
6
7 // Der Betrag der Entfernung zur aktuellen Mausposition wird berechnet.
8 // Dazu werden Beide Positionen subtrahiert, quadriert (.pow) und aus dem entstanden wert
9 // die wurzel gezogen (.sqrt).
10 // Der so entstandene wert kann als vergleichswert anderen Entfernungen genutzt werden.
11 var pointAbsMC1_x:Number = Math.abs((pointMC1.x) - (pointMouse.x));
12 var pointAbsMC1_y:Number = Math.abs((pointMC1.y) - (pointMouse.y));
13 var pointPowMC1_x:Number = Math.pow(pointAbsMC1_x, 2);
14 var pointPowMC1_y:Number = Math.pow(pointAbsMC1_y, 2);
15 var distanceMC1:Number = Math.sqrt(pointPowMC1_x + pointPowMC1_y);

```

Beispiel 15: Berechnung der Entfernung zwischen einem MovieClip (z.B. Arbeitsplatz) und den aktuellen Mauskoordinaten

⁸⁷ Vgl. [hlpadobeBD2010]

⁸⁸ Vgl. [adobelivedocSpr2010]

5.6.3 Automatische Verteilung mit Initialisierungsschlüssel

Zum Vereinfachen der manuellen Verteilung können die durch den Teamleiter geplanten Verteilungsprozesse in eigenen Funktionen erfasst und für automatisierte Prozesse eingesetzt werden. Zu diesem Zweck sind ausgewählte Informationen des Personals mit den für den Dienstplaner relevanten Arbeitsplätzen nach einem vordefinierten Schema zu vergleichen, eine resultierende Bewertungstabelle zu erstellen und anhand dieser eine automatische Personenverteilung durchzuführen.

Das Grundprinzip der automatischen Verteilung

Die die für die automatischen Verteilungsszenarien notwendigen Methoden können nach einem fast einheitlichen Grundprinzip erstellt werden:

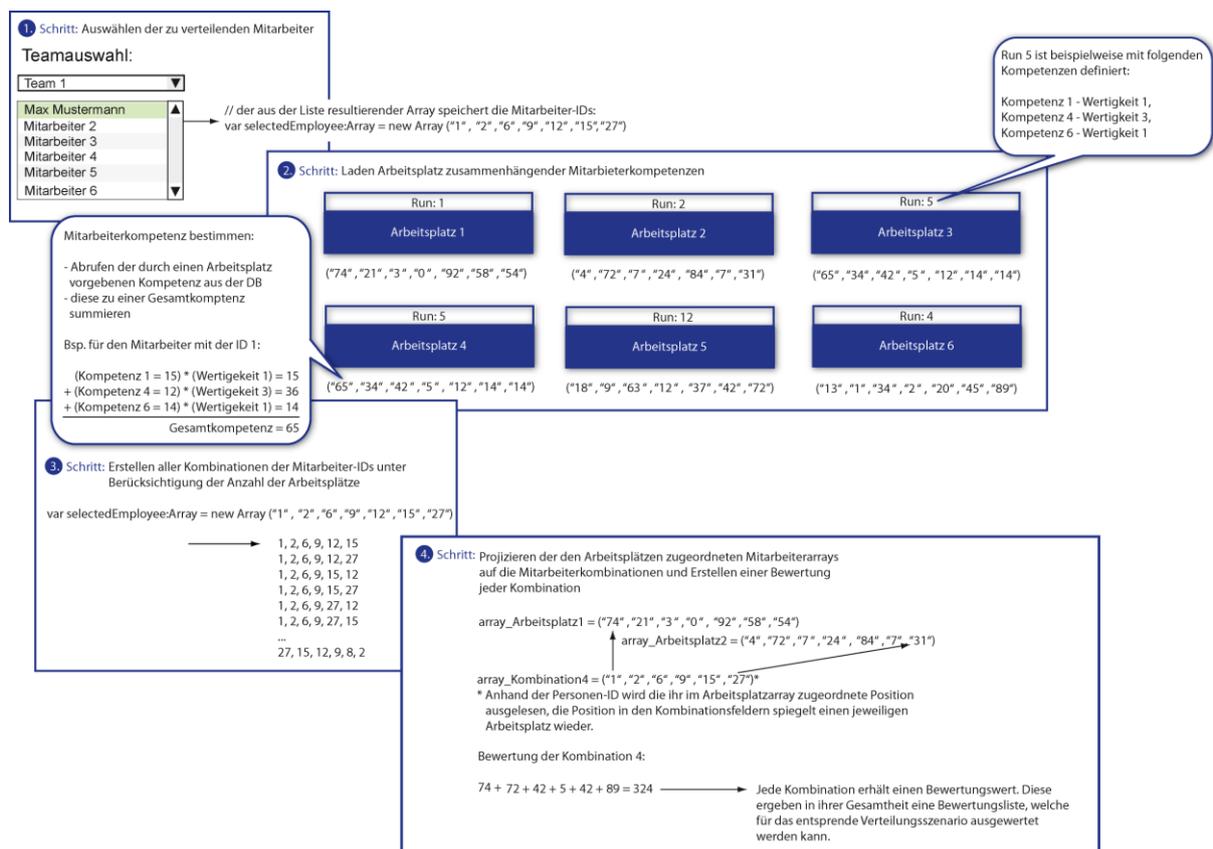


Abbildung 27: Grundprinzips der automatischen Dienstplanerstellung⁸⁹

⁸⁹ Die in der Abbildung verwendeten Werte sind frei gewählt. Sie dienen lediglich der Veranschaulichung des vorgestellten Prozesses.

1. Schritt – auswählen der zu verteilenden Arbeitsplatzressourcen: Der Teamleiter wählt den von ihm gewünschten Personenkreis und die zu besetzenden Arbeitsplätze aus. Daraus resultierend werden zwei Feldvariable (eine für die Mitarbeiter und eine für die Arbeitsplätze) angelegt. Diese werden mit eindeutig den Mitarbeitern bzw. den Arbeitsplätzen zugehörigen Kennwerten gefüllt. Im Falle der oben dargestellten Abbildung werden die Personenwerte mit den zugehörigen Benutzer-IDs initialisiert. Dies vereinfacht das Auslesen der zugehörigen Kompetenzen.

2. Schritt – Das Laden Arbeitsplatz zugehöriger Kompetenzen: Durch den Operator werden Arbeitsplätze mit zu realisierenden Arbeitsprozessen belegt⁹⁰. Diese beinhalten einzelne Tätigkeiten, welche für jeden Mitarbeiter in bewerteter Form vorliegen. Für den automatisierten Verteilungsprozess werden die einem Arbeitsplatz zugeteilten Tätigkeiten gezielt aus den Kompetenztabellen der Mitarbeiter abgerufen, summiert und die so erhaltenen Mitarbeitergesamtkompetenzen dem jeweiligen Arbeitsplatz zugeordnet. Jeder Arbeitsplatz erhält so ein eigenes Array. Die innerhalb der einem Arbeitsplatz zugeteilten Feldvariable vorhandene Reihenfolge ist mit der des Mitarbeiterarrays identisch.

3. Schritt – Erstellen möglicher Verteilungskombinationen: Um Herauszufinden, welche Gesamtverteilung der Mitarbeiter für das Verteilungsszenario am geeignetsten ist, gilt es alle anlegbaren Kombinationen zu erstellen und diese zu vergleichen.

4. Schritt – Erstellen einer Wertetabelle: Zum Vergleichen der verschiedenen Dienstplankombinationen müssen diese mit einer eigenen Wertigkeit beschrieben werden. Um dies zu erreichen werden die einzelnen den Arbeitsplätzen zugeordneten Mitarbeiterarrays auf die Verteilungsvariablen projiziert. Anhand der in diesen gespeicherten Personen-IDs wird eine zugehörige Position innerhalb der Feldvariablen der Arbeitsplätze ausgelesen. Die Position der Werte innerhalb der Verteilungsarrays bestimmt den abzufragenden Arbeitsplatz. Die Summe der so erhaltenen Werte ergibt eine zugehörige Gesamtbewertung der jeweils ausgewählten Kombination. Die Einheit aller so erhaltenen Bewertungen erzeugt eine Wertetabelle, welche zum Vergleichen der Dienstplankombinationen genutzt werden kann. Der Teamleiter hat nun die Möglichkeit eine ihm entsprechende Kombination auszuwählen.

Welche Verteilung von dem Teamleiter gewählt wird, hängt von seinen und betriebsbedingten Vorgaben ab. Die folgenden Verteilungsszenarien stellen mögliche Anordnungskriterien exemplarisch dar.

⁹⁰ Vgl. Kapitel 4.6.1 Aufgaben eines Operators

Gleichverteilung der Arbeitsplatzressourcen

Ein Teamleiter möchte, dass die an einem Arbeitsplatz geleistete Zeit für jeden Mitarbeiter möglichst gleich verteilt wird. Diesbezüglich werden die den Arbeitsplätzen zugeordneten Mitarbeiterwerte nicht durch die Personalkompetenzen sondern durch die zugehörigen Kalenderdaten erstellt. Die Gesamtsumme geleisteter Arbeitsstunden an den Arbeitsplätzen gibt eine konkrete Übersicht über den Verteilungsgrad eingesetzter Mitarbeiter. Ist diese vergleichsweise hoch, werden die Mitarbeiter sehr einseitig eingesetzt. Um eine möglichst ausgeglichene Verteilung zu bewirken muss in diesem Fall die Verteilungskombination mit der niedrigsten Gesamtsumme gewählt werden.

Nach Kompetenzen ordnen

Das Ordnen nach Kompetenzen erfolgt nach dem in Abbildung 27 dargestellten Szenario. Um eine möglichst optimale Verteilung zu bewirken, wird die Kombination mit dem höchsten Wert geladen.

5.7 Realisierung der Kalenderfunktion

Um mittel- bzw. längerfristige Termine planen zu können, wird für Personalinformationssysteme das Anlegen eines Kalenderbereichs empfohlen. Zum einen können bereits gesetzte Daten der Steuerprozesse übersichtlich dargestellt, zum anderen die Steuerprozesse blockierende Termine beantragt und organisiert werden. Die übliche Darstellung erfolgt in Form einer kalendarischen Übersicht. Diese sollte verschiedene Detaillierungsstufen darstellen und mit den jeweiligen Kalenderinformationen eines bzw. für den Teamleiter mehrerer Mitarbeiter initialisiert werden.

5.7.1 Die Tabellensätze der Kalenderübersicht

Der Kalenderbereich basiert auf einem für mittel- bzw. längerfristigen Termine angelegten Tabellensatz und den im Bereich der Steuerprozesse angelegten Mitarbeitertabellen. Jeder Eintrag wird durch eine zugehörige Antrags-ID referenziert. Die einzelnen Spalten verwalten die Person-ID eines Antragsstellers, die Terminbezeichnung, den zu belegenden Zeitraum und den aktuellen Bearbeitungsstatus einer Anfrage (vgl. Abbildung 28).

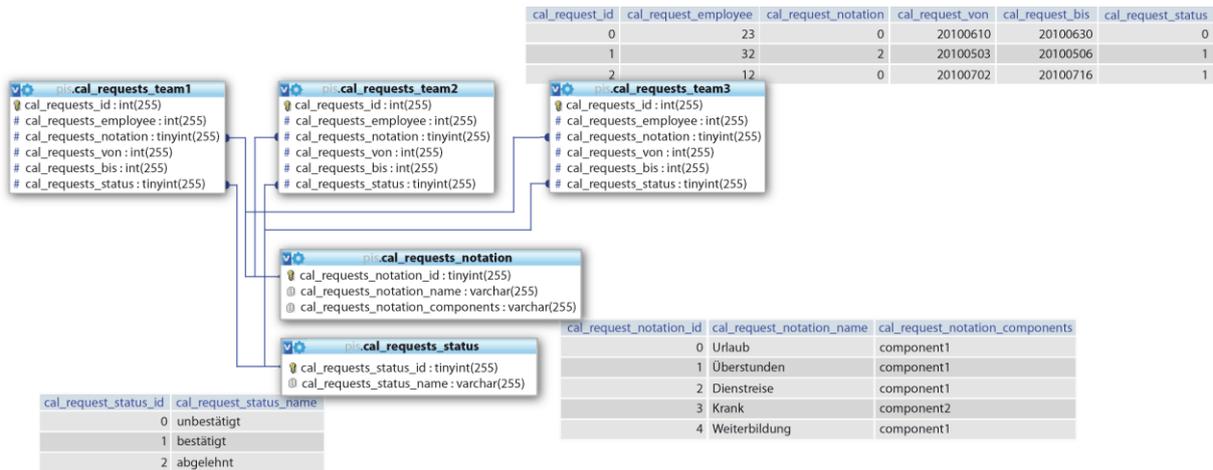


Abbildung 28: Tabellensatz des Kalenderbereichs

Zur Optimierung der Sondertermintabelle sollte diese in kleinere Tabellen aufgeteilt werden. Die Teamzugehörigkeit ist dafür sehr gut geeignet. Desweiteren werden redundante Daten in eigene Tabellen ausgelagert und durch eindeutige Werte referenziert.

5.7.2 Die Auswahlkomponente für die Datumseinstellung

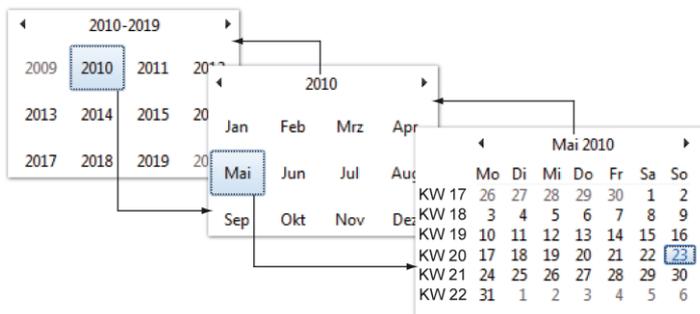


Abbildung 29: Beispielvariante einer Datumskomponente

Zum Auswählen eines Datums oder eines Zeitraumes gilt es eine eigene Komponente zu erstellen. Durch diese werden nicht nur fehlerhafte

Datumseingaben vermieden, sondern auch zusätzliche Kalenderinformationen angezeigt. Zusätzliche Angaben sind unter anderem die in einem Land vorgeschriebenen Feiertage. Eine derartige Komponente kommt in vielen Bereichen des Personalinformationssystems zum Einsatz. Sie wird beispielsweise zur Einstellung des Geburtsdatums in der Personalinformation, dem Auswählen eines zur Erstellung der Dienstpläne auszuwählenden Zeitraumes oder bei der Verteilung von Hintergrundinformationen im Operatorbereich verwendet.

Zur Erstellung der im Kalenderbereich dargestellten Übersicht oder der Kalenderauswahlkomponente wird die von Actionscript angebotene Date-Klasse verwendet. Durch diese können Datums- und Zeitinformationen aufgerufen werden. Eine erstellte Instanz dieser Klasse repräsentiert einen bestimmten Zeitpunkt, dessen Eigenschaften, wie beispielsweise Monat

(month⁹¹), Tag (day), Stunden (hours) und Sekunden (seconds), abgefragt und verändert werden können. Die Date-Klasse ermöglicht das Auslesen der auf dem Betriebssystem eingestellten Ortszeit, die Eingabe eigener Datumsangaben und automatisierte Verarbeitungsroutinen zur Datumsberechnung.

5.7.3 Das Bestimmen von Feiertagen

Um eine korrekte Terminverwaltung zu gewährleisten, ist es notwendig bestehende Feiertage zu berücksichtigen. Nur so wird unter anderem die Anzahl verbrauchter Urlaubstage korrekt berechnet oder überprüft, ob das Zuteilen der Steuerprozesse in einem ausgewählten Zeitraum möglich ist.

Die meisten in Deutschland stattfindenden Feiertage sind fest vordefiniert. Bundesland abhängige Unterschiede gilt es von vornherein in den Entwicklungsprozess einzubeziehen. Das Anlegen feststehender Feiertage kann durch folgende Methode realisiert werden:

Einige Feiertage erfolgen an jährlich unterschiedlichen Tagen. Für diese existiert eine durch Carl Friedrich Gauss (1777-1885) definierte Methode, durch welche sich der in einem gewählten Jahr stattfindende Ostersonntag berechnen lässt. Anhand dieses Datums lassen sich alle anderen variablen Feiertage Deutschlands berechnen. Die folgenden Feiertage müssen auf Grundlage des Osteralgorithmus bestimmt werden:

Aschermittwoch =	Ostersonntag – 47 Tage
Karfreitag =	Ostersonntag – 2 Tage
Christi Himmelfahrt =	Ostersonntag + 39 Tage
Pfingstsonntag =	Ostersonntag + 49 Tage
Pfingstmontag =	Ostersonntag + 50 Tage
Fronleichnam =	Ostersonntag + 60 Tage

Actionscript:

```
1 // 86400 = Gesamtzahl zu addierender Tage in Millisekunden
2 var Ostersonntag:Date = new Date();
3 var Pfingstmontag:Date = new Date(Ostersonntag.getTime() + (30 * 86400));
```

Beispiel 17: Berechnung des Pfingstmontags in Actionscript

⁹¹ Die in den Klammern aufgeführten Begriffe, sind in der AS 3.0-API die durch Date zu definierenden Befehle.

5.8 Realisierung des Operatorbereiches

Der Operatorbereich dient dazu, Arbeitszyklen zu definieren und diese zeitabhängig durch einen Operator auf die zur Verfügung stehenden Arbeitsplätze zu verteilen.

Die für diesen Bereich notwendigen Realisierungsmaßnahmen ergeben sich aus den bereits vorangegangenen Kapiteln. Der Tabellensatz des Operatorbereichs basiert auf der gleichen Grundstruktur der verwendeten Tabellen der Personalkompetenzen und der Steuerprozesse. Es werden lediglich, statt der dort eingetragenen Personen-IDs, den einzelnen Runs eindeutig zugeordnete Referenzwerte benutzt. Der Wert, der in den Personalkompetenzen den Bewertungsgrad eines Mitarbeiters darstellt, wird in diesem Bereich durch eine für die Bewertung ausschlaggebende Wertigkeit ersetzt. Ist diese null, wird die zugehörige Kompetenz als dem Run nicht zugehörig angesehen. Auch die Optimierungsmethoden des Tabellensatzes sind nahezu gleich. Statt Personengruppen können vordefinierte Einsatzbereiche zu Unterteilung kleinerer Tabellen genutzt werden.

Desweiteren können die bereits beschriebenen manuellen Zuteilungsmethoden der Mitarbeiter auf die Arbeitsplätze und die Verwaltungsbereiche der Personalinformationen auf die Verteilungsprozesse bzw. die Bearbeitungsbereiche des Operatorbereichs projiziert werden.

5.8.1 Das Drucken ausgewählter Bildschirmkomponenten

Neben den bereits detailliert in dieser Arbeit besprochenen Basisfunktionen eines PIS (das Informieren und das Steuern), unterstützen Personalinformationssysteme auch betriebsüberwachende Funktion (das Kontrollieren). So kann es unter Umständen vorkommen, dass angefertigten Dienstpläne, Mitarbeiterdaten oder die Anordnung der Hintergrundinformationen von Abteilungen oder Personen ohne Zugriff auf das PIS gefordert werden. Zu diesem Zweck sollten die dargestellten Inhalte auch in gedruckter Form verfügbar sein.

Das Zusammenstellen und das Drucken von Inhalten des PIS ermöglicht die PrintJob-Klasse. Es können sichtbare, dynamische oder externe Inhalte ausgewählt,⁹² nicht skalierte Dokumente an eine Druckvorlage angepasst und eigene Druckaufträge erstellt werden. Diesbezüglich wird ein PrintJob-Objekt erstellt, diesem eigene Eigenschaften (zum Beispiel Höhe und Breite des zu druckenden Dokuments) zugewiesen, angelegte Objekte der An-

⁹² Vgl. [hlpadobePr2010]

wendung auf das PrintJob-Objekt referenziert und eine automatische Skalierung auf den definierten Anzeigebereich bewirkt. Ein zu druckendes Objekt bindet automatisch alle zugehörigen Untercontainer in die Druckvorlage ein. Diese können auf Wunsch von dem Druckauftrag ausgeschlossen werden.

5.8.2 Daten exportieren

Das Ausdrucken eines Dokumentes kann in einigen Fällen nicht ausreichend sein. Unter Umständen möchte der Benutzer die in einem Personalinformationssystem erstellen Daten für eigene Anwendungen nutzen. Zu diesem Zweck existieren verschiedene PHP-seitige Lösungen, welche im Folgenden kurz genannt, erläutert jedoch nicht detailliert beschrieben werden.

Plattformunabhängiges Speichern von Datenbankwerten mittels XML

XML (Extensible Markup Language) ist eine Definitionssprache für Formatdateien. Mit Hilfe von XML, welches Ähnlichkeiten zu HTML aufweist, aber viel strenger reglementiert ist, lassen sich Formate definieren, die dazu dienen Datensätze zu präsentieren. XML wurde durch das W3C standardisiert. Es gibt mittlerweile viele Anwendungen, die XML als Datenaustauschformat nutzen. PHP unterstützt mehrere Funktionen, welche zum Erstellen eigener XML-Dateien mit den in einer Datenbank enthaltenen Informationen verwendet werden können.

Um eine XML-Datei in einem für Archivierungs- bzw. Weiterverarbeitungsprozesse nutzbares Format zu speichern wird XSLT verwendet. XSLT ist eine funktionale Programmiersprache zum Transformieren eines XML-Dokumentes in ein anderes auf Text basierendes Format, wie beispielsweise Textdateien (*.txt), HTML oder PDF.

Export von MySQL-Abfragen in CSV

Desweiteren ermöglicht PHP das Speichern von MySQL-Abfrageergebnissen in dem sogenannten CSV Format. Das Dateiformat CSV ist eine Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten. Die Abkürzung CSV steht dabei für Comma Separated Values weil die einzelnen Werte durch ein spezielles Trennzeichen, beispielsweise das

Komma, getrennt werden. Derartigen Dateien lassen sich in viele Datenbankanwendungen (MS-Access, MS-Excel, Oracle) aufrufen und für eigene PIS fremde Aufgaben verwenden.

5.9 Das Abmelden eines Benutzers

Möchte ein Benutzer die Arbeit mit dem PIS beenden, meldet sich dieser über ein entsprechendes Objekt ab. Die geladen Anwenderbasisdateien, welche in dieser Arbeit konzeptionell als `index.swf` bezeichnet wurde, wird entfernt und es erfolgt eine direkte Weiterleitung auf den Anmeldebereich.

An dieser Stelle zeigt sich der große Vorteil beim verwenden von PHP 5. Datenbankverbindungen werden mit dem Beenden eines PHP-Skripts automatisch geschlossen und die verwendeten Ressourcen freigegeben. Durch das Vorhandensein eines in PHP implementierten Garbage Collectors gilt dies auch für die im Personalinformationssystem verwendeten Variablen. PHP erkennt, ob eine Variable verwendet wird. Ist dies nicht der Fall wird eine automatische Ressourcenfreigabe durchgeführt.

Anders verhält es sich bei den durch Actionscript lokal gespeicherten Objekten (Shared Objects⁹³). Diese sind permanent und beim Beenden einer Actionscriptanwendung explizit zu löschen. Dies ist durchaus sinnvoll, da beim Anmelden anderer Nutzer auf dem gleichen Arbeitsplatz die vorgenommenen Einstellungen, auch auf die neu initialisierten Objekte übertragen werden. Angestellte könnten so unter Umständen verschiedene Arbeitsprozesse eines Kollegen nachvollziehen, was in den meisten Fällen nicht erwünscht ist

Actionscript:

```
1 // Funktion zum Löschen der lokalen Speicherobjekte
2 // Diese wird in das Abmeldeobjekt integriert
3 public function sharedObject_clear() {
4     so.clear();
5 }
```

Bsp 18: Löschen der Angelegten SharedObjects

⁹³ Vgl. Kapitel 5.3.6 Das Speichern von Komponentenstatements

6 Ergebnisse und Ausblick

6.1 Fazit

Im Mittelpunkt dieser Arbeit steht die Aufgabe eine einführende Übersicht zur Erstellung eines Personalinformationssystems dazustellen. Daraus resultierend werden die grundlegenden Schritte, die zur Realisierung einer solchen Anwendung notwendig sind, herausgearbeitet.

Neben dem einführenden Kapitel des Personalwesens und der Beschreibung einer prinzipiellen Herangehensweise bzw. Planung einer derartigen Anwendung werden dem Leser anhand einer konzipierten Beispielanwendung die wichtigsten Kernbereiche eines Personalinformationssystems vorgestellt. Die für die Realisierung eines PIS angelegten Kapitel beschreiben nicht nur grundlegende Technologien, sondern ergänzen diese durch zahlreiche Analysen bei der Entwicklung zu beachtender Problemstellungen, anwendungsoptimierende Gedanken und funktionelle Methoden, um den optimalen Einstieg zur Entwicklung einer solchen Software zu ermöglichen.

Das Ziel dieser Arbeit ist somit erreicht. Dem Leser sollte es möglich sein, sich einen kompakten Überblick zwecks Konzeption und Entwicklung eines Personalinformationssystems anhand dieser Arbeit aneignen zu können.

Da der Umfang solch einer Diplomarbeit nur in einem begrenzten Rahmen stattfindet, wurde diese auf die wichtigsten Kernbereiche beschränkt. Zur vollständigen Umsetzung eines Personalinformationssystems müssen die dargestellten Themen tiefgründiger erörtert und an das jeweilige Unternehmen angepasst werden. Erst dann erlangt das PIS eine Funktionalität, welche die Arbeitsvorgänge eines Unternehmens derart erleichtern, dass die Anschaffungskosten eines solchen Systems gerechtfertigt sind.

6.2 Ausblick

Erweiterte Programmmodule

Zur weiteren Unterstützung personalbedingter Arbeitsprozesse in einem Unternehmen, kann das Personalinformationssystem um zahlreiche Funktionen, welche in dieser Arbeit nicht oder nur am Rande erwähnt wurden, erweitert werden. So kann beispielsweise die beschrie-

bene Mitarbeiterauswahl bei einer zu großen Anzahl von Angestellten durch eine Suchkomponente ergänzt werden. Das heißt, ein Teamleiter oder die Mitarbeiter untereinander geben Namen oder andere Suchoptionen in eine Abfrage ein und lassen sich eine direkte Ergebnisliste anzeigen, wodurch das Erreichen einer gesuchten Information zusätzlich erleichtert wird.

Desweiteren gilt es die Funktionsweise des Personalinformationssystems beim gleichzeitigen Arbeiten mit der Software mehrerer Benutzer an verschiedenen Arbeitsplätzen zu analysieren. Geänderte Informationen sollten von allen Klienten erkannt und eine unmittelbare Aktualisierung der Arbeitsplätze veranlasst werden. Zum Erstellen sogenannter Shared-Workspaces gibt es verschiedene Lösungen, die seitens Flash realisierbar sind.

Ein weiterer Entwicklungsschwerpunkt, welcher in dieser Arbeit beschrieben aber von den Kapiteln der Konzeption und der Realisierung ausgeschlossen wurde, ist das Erstellen von CMS-Funktionen. Der Inhaber der Software sollte selbst in der Lage sein einzelne Anwendungsmodule erstellen oder bearbeiten zu können. Die vorgeschlagene Grundstruktur eines PIS und die grundlegende Funktionsweise der vorgestellten Programmkomponenten sind so angelegt, dass erweiterte CMS-Funktionen problemlos ergänzt werden können.

Entwicklung der verwendeten Technologien

Die in dieser Arbeit verwendeten Technologien und die damit verbundene Softwarearchitektur wurden von vornherein auf Flash/Actionscript, PHP und MySQL festgelegt und eine Client-Server-Architektur mit einem zugehörigen Flash-Plugin erstellt. Die daraus resultierenden Vorteile wurden im Laufe dieser Arbeit ausreichend dokumentiert. Auf Grund der Kostenpflicht, der nicht eigenständigen Erweiterbarkeit und dem Nutzen einer zusätzlichen Schnittstelle, welche Performance-technisch verzögernd wirkt, sollte genau beobachtet werden, ob Flash/Actionscript auch für zukünftige Projekte zu empfehlen ist. Spätestens bei der Einführung einer kostenfreien, Plugin-unabhängigen und mit annähernd gleichem Funktionsumfang zur Erstellung grafischer Objekte ausgestatteten Alternative, sollte auf Flash verzichtet werden. HTML5 könnte eine solche Alternative darstellen.

Anlagen

Entwurfsmuster Personalbereich - Personalinformation

PIS <small>Logged in as Max Mustermann Status: Admin Logout</small>		Personal	Steuerprozesse	Kalender	Operator
Hilfe		Personal- informationen	Personal- kompetenzen	Personal- verwaltung	

Teamauswahl:

Personalinformationen von Max Mustermann

Name: Mustermann
Vorname: Max
Titel:
Straße:
PLZ:
Ort:
Geburtsdatum:

Abteilungsnummer:
Abteilungsname:
Kostenstelle:
Beschäftigt als: Team
Team: Team 1

Eintrittsdatum:
Ende Probezeit:
Befristung bis:
Kündigungsfrist:
Voraussichtlicher Austritt:
Austrittsdatum:
Austrittsgrund:
Wöchentliche Arbeitszeit:



Max Mustermann
Mitarbeiter 2
Mitarbeiter 3
Mitarbeiter 4
Mitarbeiter 5
Mitarbeiter 6

Programmdaten ändern
neue Person
Person bearbeiten
Person löschen

Entwurfsmuster Personalbereich - Personalverwaltung

PIS <small>Logged in as Max Mustermann Status: Admin Logout</small>		Personal	Steuerprozesse	Kalender	Operator
Hilfe		Personal- informationen	Personal- kompetenzen	Personal- statistiken	Personal- verwaltung

Teamauswahl:

Personalinformationen verwalten von Max Mustermann

Name:
Vorname:
Titel:
Straße:
PLZ:
Ort:
Geburtsdatum:

Abteilungsnummer:
Abteilungsname:
Kostenstelle:
Beschäftigt als:
Team:



Eintrittsdatum:
Ende Probezeit:
Befristung bis:
Kündigungsfrist:
Voraussichtlicher Austritt:
Austrittsdatum:
Austrittsgrund:
Wöchentliche Arbeitszeit:

Max Mustermann
Mitarbeiter 2
Mitarbeiter 3
Mitarbeiter 4
Mitarbeiter 5
Mitarbeiter 6

Programmdaten ändern
neue Person
Speichern
Zurücksetzen
Person löschen

Entwurfsmuster Steuerprozesse – Personen verteilen

PIS Logged in as Max Mustermann
Status: Admin
Logout

Personal	Steuerprozesse	Kalender	Operator
----------	-----------------------	----------	----------

Hilfe

Personen verteilen	Räume verteilen
--------------------	-----------------

Teamauswahl:
Team 1

Zuordnung von Personal auf Einsatzbereich

◀ 11. ▶ ◀ August 2010 ▶ ◀ Frühschicht ▶

Übersicht der Einsatzbereiche:

Einsatzbereich A	Einsatzbereich B
Max Mustermann Run A12345	Run A12345
Einsatzbereich C	Einsatzbereich D
Run B45678	Run C12345
Einsatzbereich E	Einsatzbereich F
Run A12345	Run A12345

gesetzte Mitarbeiter ausblenden

Programmdaten ändern

Tooltip Kompetenzwerte:
Kompetenz 1: 2
Kompetenz 4: 1
Kompetenz 16: 1

Übersicht der Kompetenzdaten von Max Mustermann:

Kompetenzbereich A

Kompetenz 1	82% max. Skills
Kompetenz 2	82% max. Skills
Kompetenz 3	53% max. Skills
Kompetenz 4	82% max. Skills
Kompetenz 5	57% max. Skills

Kompetenzbereich B

Kompetenz 6	52% max. Skills
-------------	-----------------

eingeteilte Runs in der KW 32:

Mo 09.08.2010	A12345	EB A	Spät
Mo 09.08.2010	A12346	EB A	Mittel
Di 10.08.2010	A12345	EB A	Früh
Di 10.08.2010	A12346	EB A	Spät
Mi 11.08.2010	A12345	EB A	Früh

automatisch verteilen zurücksetzen

Entwurfsmuster Operatorbereich - Runverwaltung

PIS Logged in as Max Mustermann
Status: Admin
Logout

Personal	Steuerprozesse	Kalender	Operator
----------	----------------	----------	-----------------

Hilfe

Runbesetzung	Runverwaltung	Runstatistiken
--------------	----------------------	----------------

Runinformationen Runkompetenzen

Einsatzbereich wählen:
Einsatzbereich A

Runinformationen - Run A12345

Runname	A12345
Runlänge in min	120
Einsatzbereich	Einsatzbereich A
Zusätzliche Bez. 1	
Zusätzliche Bez. 2	

Run A12345
Run A12346
Run A12347
Run A12348
Run A12349
Run A12350

Programmdaten ändern

neuer Run Run speichern Run zurücksetzen Run löschen

Entwurfsmuster Operatorbereich - Runinformationen

PIS Logged in as Max Mustermann
Status: Admin
Logout.

Personal

Steuerprozesse

Kalender

Operator

Hilfe

Run-
besetzung

Run-
verwaltung

Run-
statistiken

Runinformationen

Runkompetenzen

Einsatzbereich wählen:

Einsatzbereich A

Run A12345

Run A12346

Run A12347

Run A12348

Run A12349

Run A12350

Runinformationen - Run A12345

Kompetenzbereich A

Kompetenz 1 Skill 2

Kompetenz 2 Skill 5

Kompetenz 3 Skill 1

Kompetenz 4 Skill 1

Kompetenz 5 Skill 1

Kompetenzbereich B

Kompetenz 6 Skill 1

Kompetenz 7 Skill 1

Kompetenz 8 Skill 1

Kompetenz 9 Skill 1

Kompetenz 10 Skill 1

Kompetenz 11 Skill 1

Kompetenzbereich C

Kompetenz 12 Skill 1

Kompetenz 13 Skill 1

Kompetenz 14 Skill 2

Kompetenzbereich D

Kompetenz 15 Skill 1

Kompetenz 16 Skill 1

Kompetenz 17 Skill 1

Kompetenz 18 Skill 1

Kompetenzbereich E

Kompetenz 19 Skill 1

Kompetenz 20 Skill 2

Programmdaten ändern

neuer Run

Run speichern

Run zurücksetzen

Run löschen

Entwurfsmuster Login Bereich

Benutzername

Mitarbeiter1

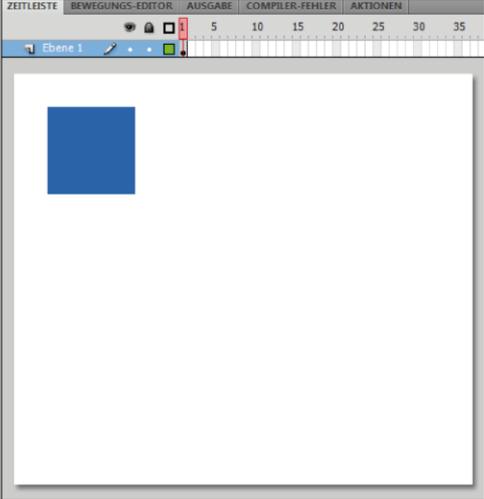
Passwort

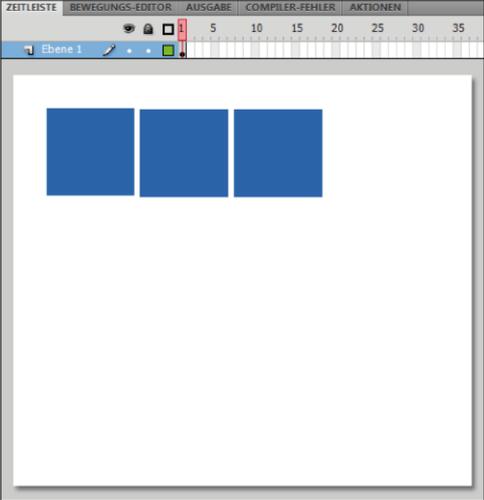
Login

Anlagen

111

Vergleichstest des verwendeten Speicherplatzes mit der Flashbühne oder Actionscript

mit Flashbühne		mit Actionscript 3.0	
		<pre>package { import flash.display.Shape; import flash.display.Sprite; public class ShapeExample extends Sprite { public function ShapeExample() { [...] } private function doDrawRect(tmp_x, tmp_y):void { var child:Shape = new Shape(); child.graphics.beginFill(0x0066CC); child.graphics.drawRect(tmp_x, tmp_y, 100, 100); child.graphics.endFill(); addChild(child); } } }</pre>	
Benötigter Speicherplatz	651 Byte		734 Byte

mit Flashbühne		mit Actionscript 3.0	
		<pre>package { import flash.display.Shape; import flash.display.Sprite; public class ShapeExample extends Sprite { public function ShapeExample() { [...] } private function doDrawRect(tmp_x, tmp_y):void { var child:Shape = new Shape(); child.graphics.beginFill(0x0066CC); child.graphics.drawRect(tmp_x, tmp_y, 100, 100); child.graphics.endFill(); addChild(child); } } }</pre>	
Benötigter Speicherplatz	891 Byte		766 Byte

Beispielcode PDO

PHP:

```
1 class Login {
2
3     // Initialisierung der Anmeldedaten
4     private $login_name;
5     private $login_password;
6
7     function __construct($login_name, $login_password) {
8
9         $this->login_name = $login_name;
10        $this->login_password = $login_password;
11
12        try {
13
14            // Herstellen einer Datenbankverbindung
15            $this->dbh = new PDO("mysql:host=localhost;dbname=pis", 'root', '');
16
17        }
18        catch (PDOException $error) {
19
20            // Datenbank wurde nicht gefunden
21            echo "Fehler beim Öffnen der Datenbank.";
22            echo "Meldung: " . $error->getMessage();
23        }
24    }
25
26    [...]
27
28 }
```


Literaturverzeichnis

- [AbtsMuelder2008]** Dietmar Abts, Wilhelm Mülder: Grundkurs Wirtschaftsinformatik: Eine kompakte und praxisorientierte Einführung, Ausgabe 6, Vieweg+Teubner Verlag, 2008, ISBN 3-8348-059-63
- [adobelivedocEv2010]** <https://www.adobe.com/livedocs/flash/9.0/ActionScriptLangRefV3/flash/events/Event.html#CHANGE>, verfügbar am 20.05.2010
- [adobelivedocSpr2010]** [http://livedocs.adobe.com/flash/9.0_de/ActionScriptLangRefV3/flash/display/Sprite.html#startDrag\(\)](http://livedocs.adobe.com/flash/9.0_de/ActionScriptLangRefV3/flash/display/Sprite.html#startDrag()) bzw. [#stopDrag\(\)](http://livedocs.adobe.com/flash/9.0_de/ActionScriptLangRefV3/flash/display/Sprite.html#stopDrag()), verfügbar am 20.05.2010
- [Brauweiler2008]** Hans-Christian Brauweiler, Unternehmensführung heute, Verlag Oldenbourg, 2008, ISBN 3-4865-825-18
- Buhl2004** Axel Buhl: Grundkurs Software-Projektmanagement, Hanser Verlag, 2004, ISBN 3-4462-262-14
- [BullingerFaehnrich1997]** Hans-Jörg Bullinger, Klaus-Peter Fähnrich: Betriebliche Informationssysteme: Grundlagen und Werkzeuge der methodischen Softwareentwicklung, Verlag Springer, 1997, ISBN 3-5406-127-42
- [BullSchustWilh2000]** Hans-Jörg Bullinger, Erwin Schuster, Stephan Wilhelm: Content Management Systeme: Auswahlstrategien, Architekturen und Produkte, Verlagsgruppe Handelsblatt GmbH, 2000, ISBN 3-7754-016-95
- [BundWirtTechn2007]** Bundesministerium für Wirtschaft und Technologie 2007, Studie: „Wertschöpfungsverluste durch nicht besetzbare Stellen beruflich Hochqualifizierter in der Bundesrepublik Deutschland“, <http://www.bmwi.de/BMWi/Redaktion/PDF/W/wertschoepfungsverluste>, verfügbar am 24.05.2010
- [DINTaschenb2262009]** DIN, DIN-Taschenbuch Nr.226: Qualitätsmanagement -

QM-Systeme und –Verfahren, 6. Auflage, Verlag BEUTH, 2009, ISBN: 3410170073

- [devmysqlcomDS2010]** <http://dev.mysql.com/doc/refman/5.1/de/data-size.html>, verfügbar am 20.05.2010
- [devmysqlcomSE2010]** <http://dev.mysql.com/doc/refman/5.1/de/storage-engines.html>, verfügbar am 20.05.2010
- [DrummHJ2008]** Hans Jürgen Drumm: Personalwirtschaft, 6., Verlag Springer, ISBN 3-5407-769-66
- [DunkelHolitschke2003]** Jürgen Dunkel, Andreas Holitschke: Softwarearchitektur für die Praxis, Verlag Springer, 2003, ISBN 3-5400-022-19
- [FeldbrBrechtHadra2008]** Rainer Feldbrügge, Barbara Brecht-Hadraschek: Prozessmanagement leicht gemacht: Geschäftsprozesse analysieren und gestalten, Ausgabe 2, Verlag Redline Wirtschaft, 2008, ISBN 3-6360-155-59
- [FruehLudewigSandm2001]** Karol Frühauf, Jochen Ludewig, Helmut Sandmayr: Software-Projektmanagement und –Qualitätssicherung, Ausgabe 4, vdf Hochschulverlag AG, 2001, ISBN 3-7281-282-28
- [GruenefLangem1991]** Hans-Günther Grünefeld u. Wigand Langemeyer: Personalinformationssystem: Entwicklung, Anwendung, Nutzen, Verlag Gabler, 1991, ISBN 3-4091-385-52
- [GummSommer2006]** Heinz-Peter Gumm, Manfred Sommer: Einführung in die Informatik, Ausgabe 7, Oldenbourg Wissenschaftsverlag, 2006, ISBN 3-4865-811-55
- [hlpadobeBD2010]** [http://help.adobe.com/de_DE/AS3LCR/Flash_10.0/flash/display/BitmapData.html#hitTest\(\)](http://help.adobe.com/de_DE/AS3LCR/Flash_10.0/flash/display/BitmapData.html#hitTest()), verfügbar am 20.05.2010
- [hlpadobeEv2010]** http://livedocs.adobe.com/flash/9.0_de/ActionScriptLang

- RefV3/flash/events/Event.html, verfügbar am 20.05.2010
- [hlpadobeCO2010]** http://help.adobe.com/de_DE/AS3LCR/Flash_10.0/fl/controls/package-detail.html detailliert dargestellt. , verfügbar am 20.05.2010
- [hlpadobeMC2010]** http://help.adobe.com/de_DE/AS3LCR/Flash_10.0/flash/display/MovieClip.html#enabled, verfügbar am 20.05.2010
- [hlpadobePr2010]** http://help.adobe.com/de_DE/AS3LCR/Flash_10.0/flash/printing/package-detail.html, verfügbar am 20.05.2010
- [hlpadobeProg2010]** http://help.adobe.com/de_DE/ActionScript/3.0/ProgrammingAS3/,
- [hlpadobePS2010]** „http://help.adobe.com/de_DE/AS3LCR/Flash_10.0/package-summary.html“, verfügbar am 20.05.2010
- [hlpadobeUC2010]** http://help.adobe.com/de_DE/ActionScript/3.0/UsingComponentsAS3/, verfügbar am 20.05.2010
- [HRAmpelCheck32007]** HR-Ampel-Check 3.0: Umfrageergebnisse zur Standortbestimmung Personalmanagement, 2007, http://www.haufe.de/Auftritte/ShopData/media/attachmentlibraries/rp/PM1008_HRAmpel_Check.pdf, verfügbar am 20.05.2010
- [Jung2006]** Hans Jung: Allgemeine Betriebswirtschaftslehre, 10., Oldenbourg Wissenschaftsverlag, 2006, ISBN 3-4865-804-93
- [Jung2008]** Hans Jung: Personalwirtschaft, 8., Oldenbourg Wissenschaftsverlag, 2008, ISBN 3-4865-870-99
- [Kamaras2003]** Endre Kamaras: Humankapital: Grund des Wachstums?, Verlag Tectum Verlag DE, ISBN 3-8288-851-87
- [Kofler2005]** Michael Kofler: MySQL 5 Open source library, Ausgabe 3, Verlag Pearson Education, 2005, ISBN 3-8273-225-37

- [Krause2005]** Jörg Krause: PHP 5 - Grundlagen und Profiwissen, Ausgabe 2, Hanser Verlag, 2005, ISBN 3-4464-033-45
- [Kruchten1999]** Philippe Kruchten: Der rational Unified Process: Eine Einführung, Verlag Pearson Education, 1999, ISBN 3-8273-154-33
- [Lassmann2006]** Wolfgang Lassmann: Wirtschaftsinformatik: Nachschlagewerk für Studium und Praxis, Verlag Springer, 2006, ISBN 3-4091-272-59
- [Muelder2000]** Mülder W., Personalinformationssysteme - Entwicklungsstand, Funktionalität und Trends, In: Wirtschaftsinformatik 42 (2000), Sonderheft, S.98 -106, ISBN 3-8348-059-63
- [MYSQL2010]** <http://www.mysql.de>, verfügbar am 20.05.2010
- [Oechsler2005]** Walter A. Oechsler: Personal und Arbeit: Grundlagen des Human Resource Management und der Arbeitgeber-arbeitnehmer-beziehungen, 8., Oldenbourg Lehr- und Handbücher der Wirtschafts- und Sozialwissenschaften, 2005, ISBN 3-4865-796-22
- [PaerliCaplaziSuter2007]** Kurt Pärli, Alexandra Caplazi, Caroline Suter: Recht gegen HIV/aids-diskriminierung im Arbeitsverhältnis: Eine rechtsvergleichende Untersuchung zur Situation in Kanada, Grossbritannien, Frankreich, Deutschland und der Schweiz, Haupt Verlag AG, 2007, ISBN 3-2580-723-02
- [PombergerPree2004]** Gustav Pomberger, Wolfgang Pree: Software Engineering: Architektur-design und Prozessorientierung, Ausgabe 3, Hanser Verlag, 2004, ISBN 3-4462-242-97
- [Rohweder1996]** Dirk Rohweder: Band 9 von Unternehmensführung u. Logistik: Informationstechnologie und Auftragsabwicklung, Erich Schmidt Verlag GmbH, 1996, ISBN 3-5030-391-12
- [StatBundAmt2004]** Statistisches Bundesamt Deutschland:
<http://www.destatis.de/jetspeed/portal/cms/Sites/destatis/lnetnet/DE/Presse/pk/2006/Arbeitskostenerhebung/Arbeits>

kostenerhebung__06,templateId=renderPrint.psml, verfügbar am 24.05.2010

- [Schroeder2009]** Olaf Schröder: Gestaltungsfragen eines Personalinformationssystems, Verlag Grin, 2009, ISBN 3-6403-466-7X
- [SchmidtHompelTen2010]** Thorsten Schmidt, Michael Hompel, Ten: Warehouse Management: Organisation und Steuerung von Lager- und Kommissioniersystemen, Verlag Springer, 2010, ISBN 3-6420-318-46
- [StaBundesamtD2004]** Statistisches Bundesamt Deutschland, <http://www.destatis.de>, verfügbar am 20.05.2010
- [Taborsky2008]** Stephan Alexander Taborsky: Planung und Analyse eines idealen Rechtsinformationssystems, Verlag BoD – Books on Demand, 2008, ISBN 3-8366-603-93
- [TraegerVolk2002]** Dirk H Traeger: Andreas Volk, LAN: Praxis lokaler Netze, Ausgabe 4, Vieweg+Teubner Verlag, 2002, ISBN 3-5193-618-92
- [TurnerPysterPennotti2009]** Systems Conference, 2009 3rd Annual IEEE Volume, Issue 2009: <http://ewh.ieee.org/soc/pels/pdf/23npels03-forweb.pdf>, verfügbar am 20.05.2010
- [vmodell972010]** <http://v-modell.iabg.de>, Rubrik V-Modell 97, verfügbar am 20.05.2010
- [vmodellXT2010]** <http://v-modell.iabg.de>, Rubrik V-Modell XT, verfügbar am 20.05.2010
- [Wolfgarten2004]** Sebastian Wolfgarten, Apache Webserver 2, Ausgabe2, Verlag Pearson Education, 2004, ISBN 3-8273-211-82
- [Zapp2006]** Ralf-Michael Zapp: Systematische Personalauswahl und ihre rechtlichen Rahmenbedingungen, Martin Meidenbauer Verlag, 2006, ISBN 3-8997-505-94

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, den 20.05.2010