

TriplePlay

**Ein Softwareprojekt im Rahmen der Lehrveranstaltung SWP1 und SWP2 im Studiengang
Medieninformatik im Wintersemester 2004/05 und im Sommersemester 2005 an der TFH.**

Siamak Haschemi (s718113) und Philipp Müller (s717803)

8. Juli 2005

Inhaltsverzeichnis

1	Projektbeschreibung	5
1.1	Der Kunde	5
1.2	Ursprüngliches Kunden-Szenario	5
1.3	Das Videotext-Szenario	7
1.3.1	Videotext im DVB-Strom	7
2	Realisierung	9
2.1	eXtreme Programming	9
2.1.1	Pair Programming	9
2.1.2	XPlanner als Projektplanungsplattform	9
2.2	CVS	9
2.3	Iterationen	10
2.3.1	30.10.2004 bis 05.11.2004	10
2.3.1.1	Standardseite der IP-STB	10
2.3.1.2	Browsertest	10
2.3.2	05.11.2004 bis 26.11.2004	11
2.3.2.1	DVBtext	11
2.3.3	29.11.2004 bis 01.12.2004	12
2.3.3.1	Der Asterisk Telefonserver	12
2.3.4	03.12.2004 bis 22.12.2004	13
2.3.5	28.12.2004 bis 07.01.2004	14
2.4	Serverstruktur	16
2.5	UML	16
3	Frameworks	21
3.1	Spring Framework	21
3.1.1	Das Spring Framework	21
3.1.2	Konfiguration	21
3.2	JavaServer Faces	22
3.2.1	Exception Framework	22
3.2.2	UI-Komponenten	22
3.3	Hibernate	22
3.3.1	Hibernate - Relational Persistence For Idiomatic Java	22
3.3.2	Datenbankschema	23
3.3.3	xDoclet - Generierung	23
3.3.4	SchemaExport	24
3.3.5	Fazit	24
4	Oberflächentestprotokoll	27
4.1	login.jsp	27
4.1.1	Benutzername	27
4.1.2	Passwort	27
4.1.3	Details	27
4.2	details.jsp	27
4.2.1	Benutzername	27
4.3	start/start.jsp	28
4.3.1	Benutzername	28
4.3.1.1	Logout	28
4.4	phone/contacts.jsp	28

5 Programmierrichtlinien	29
5.1 Allgemein	29
5.2 Sprache	29
5.3 Kommentare	29
5.4 Deklarationen	29
5.4.1 Konstanten, Variablen, Parameter	29
5.4.2 Methoden	30
5.4.3 Pakete	30
5.5 Einrückungen, Whitepaces und Klammerung	30
5.6 TODOs	30
6 Fazit	33
6.1 Das Projekt	33
6.2 Erfahrungen mit Extreme Programming	33
6.3 Beurteilung der Lehrveranstaltung	34
A Verwendete Programme	35
Literaturverzeichnis	39
Glossar	41

Kapitel 1

Projektbeschreibung

1.1 Der Kunde

Das Projekt TriplePlay wurde für einen externen Kunden realisiert. Es handelt sich dabei um die Firma *digicast GmbH*.¹

1.2 Ursprüngliches Kunden-Szenario

Ein Betreiber eines Kreuzfahrtschiffes möchte seinen Gästen Telefonieren zu günstigen Kosten, überall in der Welt den heimatischen Fernsehsender und einen schnellen Internetzugang anbieten. Um die Kosten für die Gäste transparent zu halten, sollen die Abrechnungsdaten (TV, Telefon und Internet) auf dem TV-Gerät einsehbar sein. (Annahme: Das Schiff verfügt über eine sehr schnelle Internet Anbindung.) Dazu wird zu jedem Fernseher auf dem Schiff jeweils eine IP-STB installiert.

Aufgeteilt wird das Projekt in zwei logische Einheiten:

- Die Erste Einheit bildet das gesamte Schiff und alle darin enthaltenen Teilsysteme: Hierzu gehören die IP-STBen sowie ein zentraler Server, auf dem u.A. die Kundenverwaltung und Abrechnung stattfinden.
- Die zweite Einheit bildet ein in der Anbieterfirma installiertes System. Hierzu gehören mehrere PCs, die TV und Telefon zur Verfügung stellen.

IP-TV: In der Firma stehen gängige PCs mit jeweils vier DVB-S Karten für den Empfang von Satellitenfernsehen. Jede einzelne Karte verteilt das vom Satelliten gelieferte Bild- und Tonmaterial in ein internes Netzwerk. (Hierbei werden sog. „Transport-Streams“ als Multicast zu einem multicastfähigem Switch gestreamt.)

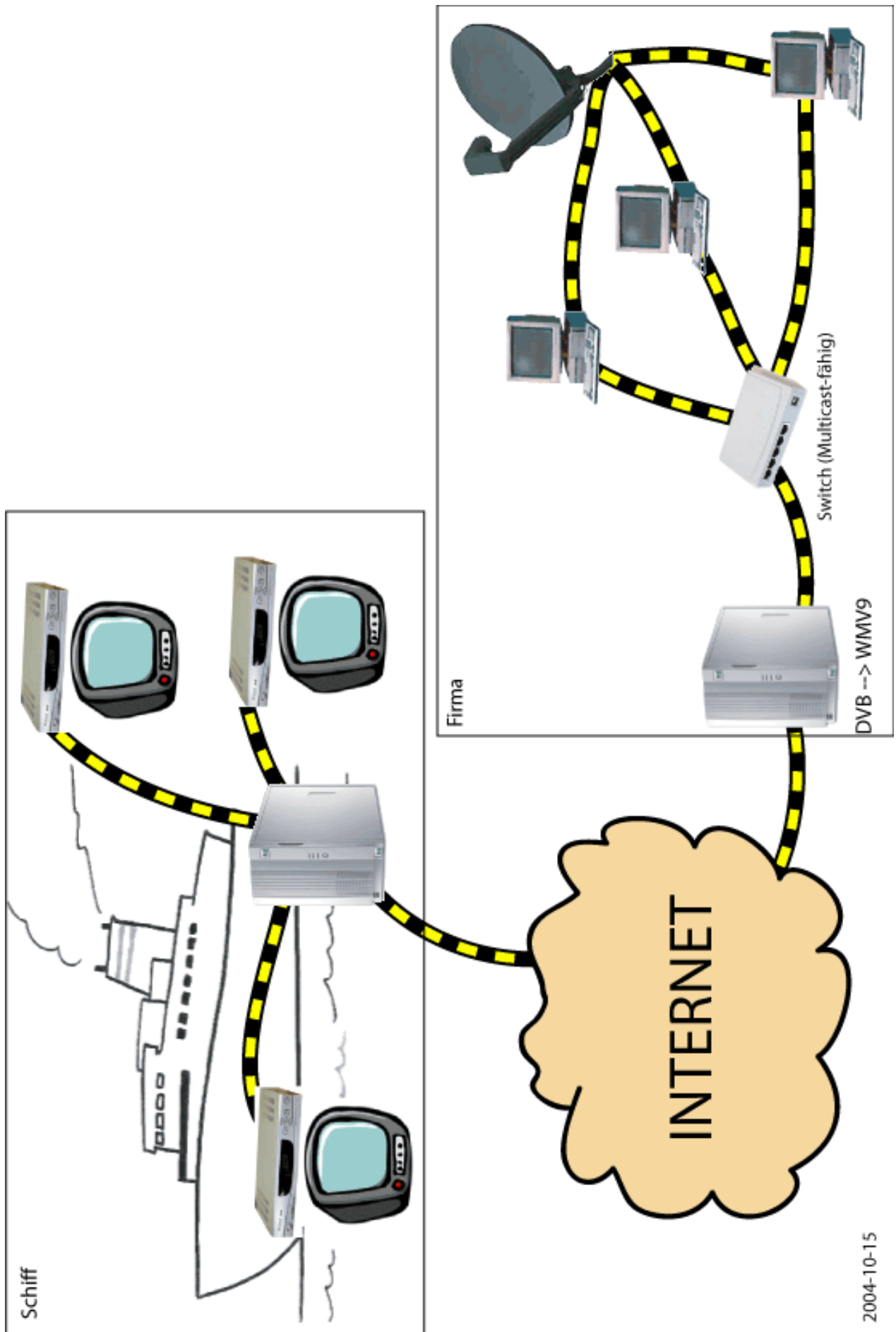
Auf dem Schiff werden Anfragen von Benutzern (z.B. soll auf „Pro7“ gewechselt werden) angenommen und der entsprechende Stream zur Weiterleitung zum Schiff vorbereitet.

Bei der Vorbereitung wird aus dem MPEG2-Format der DVB-Streams durch Echtzeitencodierung das WMV9-Format von Microsoft erzeugt. Dieses ist besonders für Streaming über das Internet geeignet, da bei geringen Datenraten gute Bild-Ton-Ergebnisse zu erzielen sind.

IP-Telefon: Mit Hilfe der SIP Technologie können Nutzer kostengünstig über das Internet telefonieren. Hierzu werden IP-Telefone installiert. Anrufe und Voice-Mails (vielleicht später sogar MMS, SMS) werden auf dem Fernseher angezeigt.

Internet: Das breitbandige Internet ermöglicht es, zu Surfen und Mails zu erhalten. Die Mails werden am TV Gerät angezeigt und können auch dort beantwortet werden.

¹Ihren Tätigkeitsbereich beschreibt der Kunde selbst wie folgt: Unter dem Claim Intelligenter Fernsehen unterstützt digicast Anbieter der Medien- und Kommunikationsbranche, Breitband-Medien-Strategien umzusetzen und Inhalte auf digitalen Netzen und Endgeräten zu vermarkten. Dabei stellt digicast die Beratung der technischen Strategie, die Ausarbeitung von tragfähigen Business-Modellen wie die Entwicklung von innovativen technischen Lösungen in den Vordergrund.



1.3 Das Videotext-Szenario

Aufgrund der Tatsache, dass nach dem Ende der Gruppenfindungsphase nur drei und kurze Zeit später nur noch zwei Studenten an diesem Projekt beteiligt waren, wurde die Anforderung des Kunden entsprechend umgestaltet. Der Umfang wurde drastisch verringert, so dass der gesamte Teil des WMV9 Streamings und das Kundenabrechnungssystem ausser Acht gelassen wurde. Statt dessen wurde der Wunsch formuliert, über die STB Zugang zu aktuellen Videotext Seiten zu haben. Der Endverbraucher soll über die Fernbedienung durch die Seiten von mehreren Fernsehsendern navigieren können. Da die STB internetfähig ist und über einen Web-Browser verfügt, sollen die Seiten über das WWW zur Verfügung gestellt werden.

Nach Auswahl eines Senders wird die Indexseite (100) angezeigt. Der Benutzer hat die Möglichkeit, durch Eingabe einer Zahl zwischen 100 und 899 zu einer entsprechenden Seite zu gelangen. Beinhaltet eine Seite weitere Unterseiten, so sollen diese angezeigt werden und auswählbar sein. Zudem ist ein Mehrwert des internetbasierten Videotextes, dass Verweise innerhalb einer Seite auf andere Seiten ebenfalls selektierbar sind. Dieses Feature ist auf vielen herkömmlichen Fernsehgeräten mit Videotextunterstützung nicht gegeben.

1.3.1 Videotext im DVB-Strom

Der Transport-Stream im DVB Signal beinhaltet neben Video- und Audio-Informationen u.A. auch kodierte Videotext Signale. Diese sind für einen bestimmten Sender über entsprechende PIDs (genauer: TTPID) auslesbar.

Eine Videotextseite kann neben reinen Text verschiedene Elemente und Steuer-Information beinhalten. Dazu zählen z.B.:

- Graphische Elemente (ca. 65 verschiedene Blocksymbole)
- Farbe für Vordergrund und Hintergrund
- Anweisungen für doppelte Zeilenhöhe
- Blinkende Zeichen
- Versteckte Informationen, die auf Wunsch sichtbar gemacht werden können

Eine Auflistung aller Steuerzeichen ist unter [\[vtxCtrlChars\]](#) zu finden.

Abbildung 1.1 Aufbau einer Videotextseite

	Status	Kopfzeile (Seitennr., Sender)	Uhrzeit
Zeile 0	7 Zeichen	1	24 Zeichen
Zeile 1			8 Zeichen
.			
.			
.			
Zeile 23			
Zeile 24	Erweiterungszeile für Spezialfunktionen		
Zeile 25	10 Bytes	14 Bytes	
	Seiten-Infos	unbenutzt	

Kapitel 2

Realisierung

2.1 eXtreme Programming

Das Projekt sollte nach Kundenwunsch möglichst mit der Vorgehensweise „eXtreme Programming (XP)“ durchgeführt werden.

eXtreme Programming ist eine relativ neue Vorgehensweise in der Softwaretechnik. Dabei wird auf einen strikten Anforderungskatalog des Kunden verzichtet, dafür werden auch Kundenwünsche berücksichtigt, die sich noch während der Softwareentwicklung ergeben. Statt des klassischen Wasserfallmodells durchläuft der Entwicklungsprozess immer wieder in kurzen Zyklen (typischerweise ein bis zwei Wochen) sämtliche Disziplinen der klassischen Softwareentwicklung (Anforderungsanalyse, Design, Implementierung, Test). Nur die im aktuellen Iterationsschritt benötigten Merkmale werden implementiert.

Die Methode hat die Erfahrung zum Hintergrund, dass der Kunde die wirklichen Anforderungen zum Projektbeginn meist noch nicht komplett kennt. Er fordert Features, die er nicht braucht und vergisst solche, die benötigt werden.

2.1.1 Pair Programming

Ein Anliegen der XP-Methode ist es, das Know-How über den gesamten Quellcode möglichst auf jeden Entwickler zu verteilen. Es soll keine Spezialisten für bestimmte Abschnitte geben. Vielmehr sollen Situationen vermieden werden, bei denen beispielsweise durch Krankheit eines Programmierers der Ansprechpartner fehlt oder schlechter Code produziert wird, der nicht mindestens durch einen zweiten Entwickler geprüft wurde.

Refaktorisierung von Quellcode, der nicht aus eigener Hand stammt, ist bei XP nicht nur Recht sondern vielmehr eine Pflicht!

2.1.2 XPlanner als Projektplanungsplattform

Um die Kundenwünsche konkret vom Umfang und Zeitbedarf planen zu können, haben wir das Tool XPlanner eingesetzt. Diese Web-Applikation eignet sich vorallem für Projekte, die mittels „eXtreme Programming“ realisiert werden. Es unterstützt die Gliederung in Iterationszyklen und bietet verschiedene Methoden der tabellarischen und graphischen Auswertung über den Entwicklungsfortschritt und der benötigten Zeit.

2.2 CVS

Um die Sicherheit eines funktionierenden Systems besonders bei der Methode des „eXtreme Programmings“ zu gewährleisten, wo Teilsysteme bereits vor der endgültigen Fertigstellung vom Kunden in Betrieb genommen werden, ist es um so wichtiger, auf eventuelle frühere Versionen des Programmcodes zurückgreifen zu können. Um Designentscheidungen, die sich im agilen Prozess der XP-Entwicklung als ineffizient erweisen, einfacher verwerfen zu können, ist ein Versionskontrollsystem wie CVS¹ äusserst wertvoll.

¹Concurrent Version System

Desweiteren ist dadurch eine gute Backup-Möglichkeit und einfacheres verteiltes Arbeiten möglich. Letzteres ist jedoch gerade aufgrund des „Pair Programmings“ eher zweitrangig.

2.3 Iterationen

Die Entstehung des Softwaresystems war ein Prozess, dessen Ziel zu Beginn nicht eindeutig vorhersehbar war. Einzelne Teilaufgaben wurden gemäß dem „eXtreme Programming“-Ansatz mit dem Kunden in meist zweiwöchigen Meetings in sog. „Stories“ verfasst und Priorisiert. Dabei standen beim Kunden hauptsächlich zwei Überlegungen im Vordergrund:

- Zeitlicher Aufwand der Umsetzung
- Eignung für Demonstrationszwecke für potentielle Kunden

Bei jedem Meeting wurde ein Protokoll angefertigt. Stories wurden direkt in XPlanner eingetragen, damit sowohl der Kunde als auch die Studenten und der Dozent jederzeit einen Überblick über die Entwicklung und die aktuellen Aufgaben machen konnten.

Die folgenden Daten der Iterationszyklen geben die im Meeting angesetzte Zeiten zur Bearbeitung der Stories an. Darüber hinaus wurden während und zwischen den Stories bis zuletzt Refaktoriesierungsarbeiten getätigt.

2.3.1 30.10.2004 bis 05.11.2004

Die erste Story bestand darin, ein Gefühl für die STB zu entwickeln. Das Ergebnis der Researchaufgaben hatte folgendes über die Box ergeben:

2.3.1.1 Standardseite der IP-STB

Durch das Sniffer Tool Ethereal wurde ermittelt, dass sich die STB zu Beginn Verbindung zu der Webseite <<http://www.insidedvd.com/aeondigital/diva/WorldNetwork.html>> aufnimmt.

2.3.1.2 Browsertest

Der Netscape-Browser der STB unterstützt folgende Formate und Technologien:

- WMV9 Videostreams
- WMA Audiostreams
- Die Bildformate GIF und JPEG
- Macromedia Flash 5.0
- CSS 1.0
- JavaScript 1.3

Zudem enthält die Box nur eine serifenlose Standardschriftart, die *nicht* monospaced ist.

Im „Browsermodus“ entstehen oben und unten schwarze Balken, was die maximal darstellbare Bildgröße auf 638x400 Pixel beschränkt.

Iteration 01 (2004-10-30 Zu 2004-11-05) [id=16]

TriplePlay
01

Ziel dieser Iteration ist es, ein Gefühl für die IP-STB zu bekommen.

Stunden: Schätzung 14,0, Verbraucht 13,2, Verbleibend 0,0

User Story	ID	!	Cust.	Progress	Est.	Act.	Rem.	Tasks	Tracker	Actions
Demo Seite mit drei WMV/WMA-Links erstellen	67	1		<div style="width: 100%; height: 10px; background-color: green;"></div>	4,0	4,0		2		
Standardseite der IP-STB ermitteln	77	1		<div style="width: 100%; height: 10px; background-color: green;"></div>	3,0	3,0		1		
Video/Audio Formate/Bitraten/Auflösungen der IP-STB ermitteln	75	2		<div style="width: 100%; height: 10px; background-color: green;"></div>	0,8	0,8		1		
Browser des IP-STBs analysieren	69	3		<div style="width: 100%; height: 10px; background-color: green;"></div>	2,4	2,4		2		
Links der Standardseite analysieren	101	3		<div style="width: 100%; height: 10px; background-color: green;"></div>	1,0	1,0		2		
Maximale darstellbare Auflösung	71	3		<div style="width: 100%; height: 10px; background-color: green;"></div>	1,0	1,0		1		
Media Server unter Linux	73	4		<div style="width: 100%; height: 10px; background-color: green;"></div>	1,0	1,0		1		

2.3.2 05.11.2004 bis 26.11.2004

Alle Tasks der Iteration 02 (2004-11-05 Zu 2004-11-26) [id=148]



Stunden: Schätzung 119,8, Verbraucht 116,8, Verbleibend 2,0

Story	!	Task	Empfänger	Geschätzt	Verbraucht	Verbleibend	Aktionen
HTML-Teletext aus DVB-Daten	1	Arbeit am Linux - Server		10,5	10,5	0,0	
HTML-Teletext aus DVB-Daten	1	Benchmark: GIF-Erzeugung		4,3	4,3	0,0	
HTML-Teletext aus DVB-Daten	1	DVB-T Server		7,8	7,8	0,0	
HTML-Teletext aus DVB-Daten	1	dvbtext & vtx2ascii Tool-Analyse		4,5	4,5	0,0	
HTML-Teletext aus DVB-Daten	1	Research Videotext (VTX)-Format		6,4	6,4	0,0	
HTML-Teletext aus DVB-Daten	1	Teletext Font		4,0	3,0	0,0	
HTML-Teletext aus DVB-Daten	1	Teletext Link-Parser		1,0	1,0	0,0	
HTML-Teletext aus DVB-Daten	1	Videotext Darstellung & Navigation		39,5	39,5	0,0	
HTML-Teletext aus DVB-Daten	1	VTX-Dekodierer Grundgerüst		5,0	5,0	0,0	
HTML-Teletext aus DVB-Daten	1	VTX-Implementation		34,8	34,8	0,0	

Abgeschlossen Nicht zugewiesen Nicht geschätzt

Notiz:

Betreff: Herr Wangas Austritt aus TriplePlay **Autor:** Haschemi **Datum:** 2004-11-13 13:35

Herr Wanga hat sich am 12.11.2004 entschieden, aus dem Projekt aus persönlichen Gründen auszutreten.

Die Story wurde wie folgt vom Kunden formuliert:

Ziel ist es, Teletext aus DVB-T (optional DVB-S) Signalen zu extrahieren. Die aktuellsten Teletextdaten werden in eine SQL-Datenbank importiert. Besonders muß auf Synchronisation geachtet werden. Die Verweise auf andere Teletext-Seiten innerhalb einer Seite sollten korrekte HTML-Links sein. Es gibt ein bestehendes Tool welches einige Grundfunktionen bereitstellt, die man nutzen könnte (DVBtext).

2.3.2.1 DVBtext

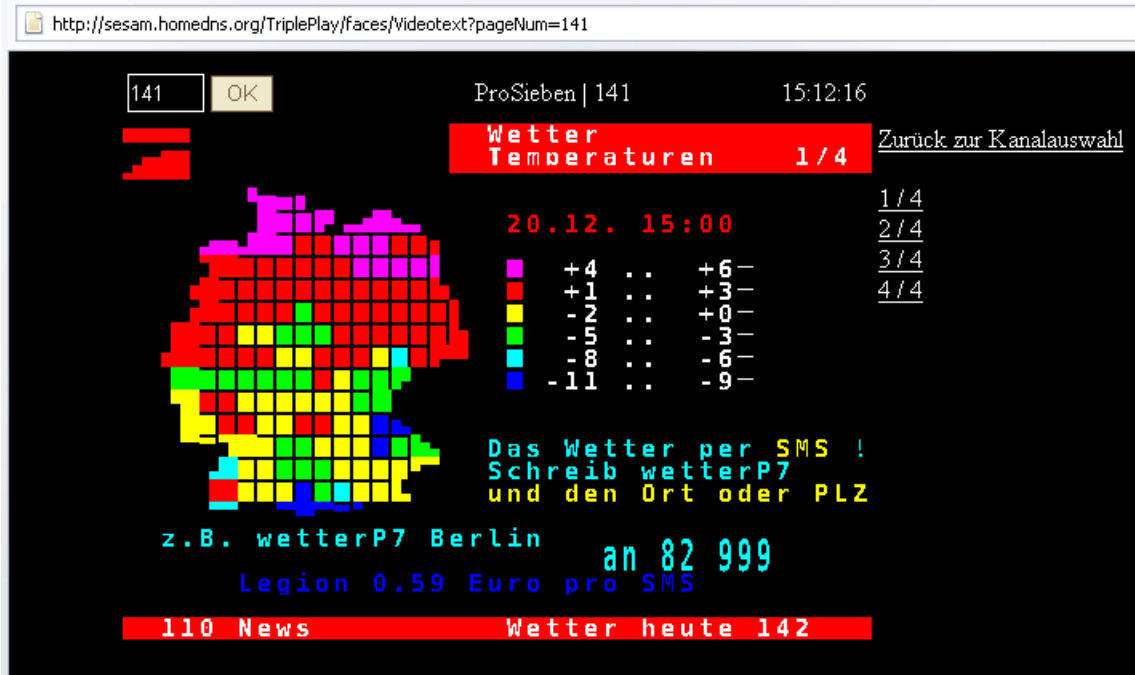
Das Tool DVBtext ist ein OpenSource Programm, welches Videotextseiten von bis zu acht Programmen (siehe **Abbildung 2.1**) eines Transponders über deren TTPIDs herausfiltert und in Dateien piped, welche die Namensform SEITE_UNTERSEITE.vtx haben und in Unterverzeichnisse der Form SENDENAME abgelegt werden. Die SEITE ist eine Nummer zwischen 100 und 899, die UNTERSEITE kann zwischen 00 und 99 liegen.

Ein Beispiel für eine Videotext-Seite ist in **Abbildung 2.2** zu sehen.

Abbildung 2.1 Kanalauswahl



Abbildung 2.2 Videotext



2.3.3 29.11.2004 bis 01.12.2004

Alle Tasks der Iteration 03 (2004-11-29 Zu 2004-12-01) [id=277]



Stunden: Schätzung 37,3, Verbraucht 37,2, Verbleibend 0,0

Story	!	Task	Empfänger	Geschätzt	Verbraucht	Verbleibend	Aktionen
Konferenzschaltung	4	Weboberfläche Konferenzerstellung		20,8	20,8	0,0	🗑️ 📝 ✖️
Konferenzschaltung	4	Weboberfläche Nummernverwaltung		8,0	8,0	0,0	🗑️ 📝 ✖️
Konferenzschaltung	4	Überarbeitung des Exception-Handlings		8,5	8,5	0,0	🗑️ 📝 ✖️

Abgeschlossen Nicht zugewiesen Nicht geschätzt

Die Story wurde wie folgt vom Kunden formuliert:

Es soll möglich sein, von der STB aus eine Konferenzschaltung einzuleiten. Dazu werden zwei Weboberflächen erstellt:

- Es soll eine Weboberfläche erstellt werden, in der man einer Person eine Telefonnummer zuordnen kann. Die Eintragung der Daten muss nicht von der Settop-Box aus geschehen.
- In der zweiten Oberfläche wählt man bis zu 4 Personen für eine Konferenzschaltung aus. Diese Oberfläche muss über die STB bedienbar sein. Es wird davon ausgegangen, dass die Rufnummern über ISDN erreichbar sind.

2.3.3.1 Der Asterisk Telefonserver

Der Asterisk Telefonserver bietet eine Konferenzfunktion für eine unbegrenzte Anzahl von Teilnehmern an. Benutzer können sowohl über ISDN als auch über VoIP eingeladen werden. Diese Funktionalität wurde über eine HTML-Seite verfügbar gemacht. Es ist möglich, Kontakte anzulegen (siehe **Abbildung 2.3**) und diese in eine Konferenzschaltung einzuladen (siehe **Abbildung 2.4**). Für jede Nummer kann zudem angegeben werden, ob sie über ISDN oder VoIP gewählt werden soll.

Abbildung 2.3 Kontakte

http://sesam.homedns.org/TriplePlay/faces/pages/phone/contacts.jsp

digicast

Name

Nummer

Kontakte

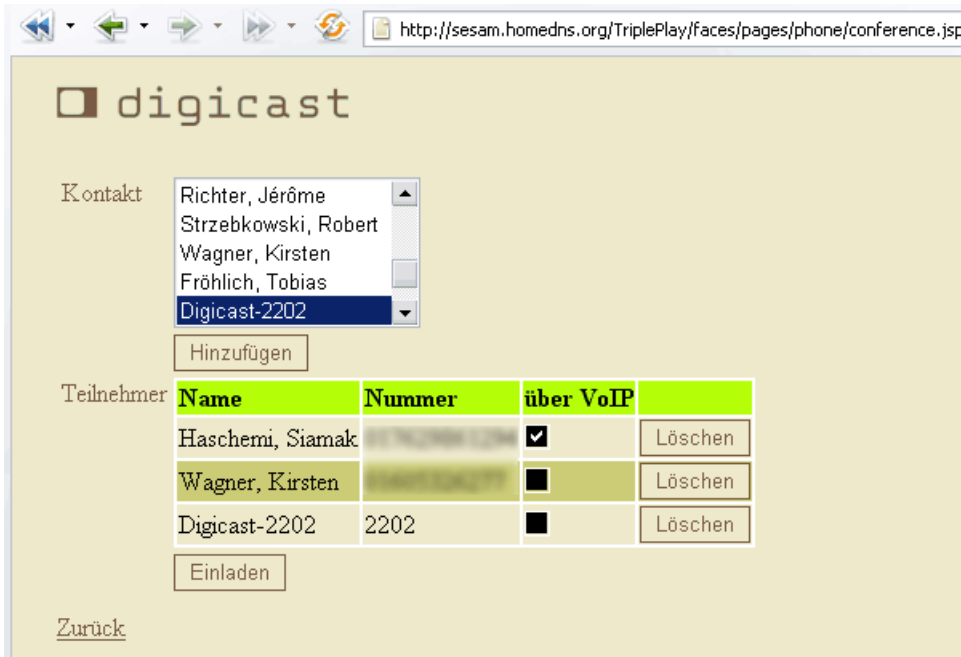
Name	Nummer	
Brandt, Thoralf	01723941782	<input type="button" value="Löschen"/>
Buchholz	01723941782	<input type="button" value="Löschen"/>
Foth, Gerd-A.	01723941782	<input type="button" value="Löschen"/>
Fuchs, Sylvio	01723941782	<input type="button" value="Löschen"/>
Hartenthaler, Hermann	01723941782	<input type="button" value="Löschen"/>
Horbat, Manfred	01723941782	<input type="button" value="Löschen"/>
Goldhammer, Klaus	01723941782	<input type="button" value="Löschen"/>
Langer	01723941782	<input type="button" value="Löschen"/>
Müller-Jones, Kay	01723941782	<input type="button" value="Löschen"/>
Richter, Jérôme	01723941782	<input type="button" value="Löschen"/>
Strzebkowski, Robert	01723941782	<input type="button" value="Löschen"/>
Wagner, Kirsten	01723941782	<input type="button" value="Löschen"/>
Fröhlich, Tobias	01723941782	<input type="button" value="Löschen"/>
Digicast-2202	01723941782	<input type="button" value="Löschen"/>
Digicast-3301	01723941782	<input type="button" value="Löschen"/>
Haschemi, Siamak	01723941782	<input type="button" value="Löschen"/>

2.3.4 03.12.2004 bis 22.12.2004

Die Story wurde wie folgt vom Kunden formuliert:

Videotextdaten sollen textuell verglichen werden, um unnötige Konvertierungen zu Bildern und Zugriffe auf die Datenbank zu vermeiden. Dazu soll DVbtext so abgeändert werden, dass die Paritätsprüfung und ein textueller Vergleich dort stattfindet. Desweiteren soll eine Datei nur dann persistent gemacht werden, wenn der Inhalt der Videotextseite sich geändert hat. Dazu muss die erste Zeile ignoriert werden, da dort die Uhrzeit angegeben ist.

Abbildung 2.4 Konferenzschaltung



Alle Tasks der Iteration 04 (2004-12-03 Zu 2004-12-22) [id=307]



Stunden: Schätzung 74,3, Verbraucht 71,3, Verbleibend 0,0

Story	!	Task	Empfänger	Geschätzt	Verbraucht	Verbleibend	Aktionen
TriplePlay-Portal	2	ErrorPage		8,5	8,5	0,0	
TriplePlay-Portal	2	Login		3,0	3,0	0,0	
TriplePlay-Portal	2	Refaktorisierung		6,5	6,5	0,0	
Videotext-Optimierung	4	Kontinuierliche Uhrzeit auf Videotextseite		7,0	4,0	0,0	
Videotext-Optimierung	4	Refaktorisierungen		17,2	17,2	0,0	
Videotext-Optimierung	4	Server Administration		10,8	10,8	0,0	
Videotext-Optimierung	4	Textueller Vergleich von Videotextseiten		3,5	3,5	0,0	
Videotext-Optimierung	4	Unterstützung mehrerer Sender		17,8	17,8	0,0	

Abgeschlossen Nicht zugewiesen Nicht geschätzt

2.3.5 28.12.2004 bis 07.01.2004

Die Story wurde wie folgt vom Kunden formuliert:

Ein externer Kunde greift auf einen Service innerhalb der Firma zu. Sollte der Zugriff scheitern, hat er die Möglichkeit, eine andere URL zu wählen, wo der Zugriff zuverlässiger ist, die Daten jedoch älter sein dürfen.

- Die STB-Funktionalität soll für externe Kunden über einen Browser zugänglich gemacht werden. Der Zugang soll durch Benutzername und Passwort geschützt werden.
- Es soll möglich sein, neue Benutzer anzulegen.
- Jeder Benutzer soll in der Lage sein, seine eigenen Kontakte anlegen zu können.
- Jeder Einloggsvorgang soll in einer Datei mit Datum erfasst werden.
- Die Uhrzeit auf den Videotextseiten soll auch im Web korrekt dargestellt werden.
- Alle Videotext-relevanten Daten sollen auf einem externen Server gespiegelt werden. Dabei sollen möglichst nur aktualisierte Daten transferiert werden.
- Der Telefon-Server soll für VoIP Unterstützung konfiguriert werden (SipGate?).
- Der Telefon-Server soll mit einer zweiten ISDN-Leitung arbeiten können.

Abbildung 2.5 Hauptmenü



Alle Tasks der Iteration 05 (2004-12-28 Zu 2005-01-07) [id=375]

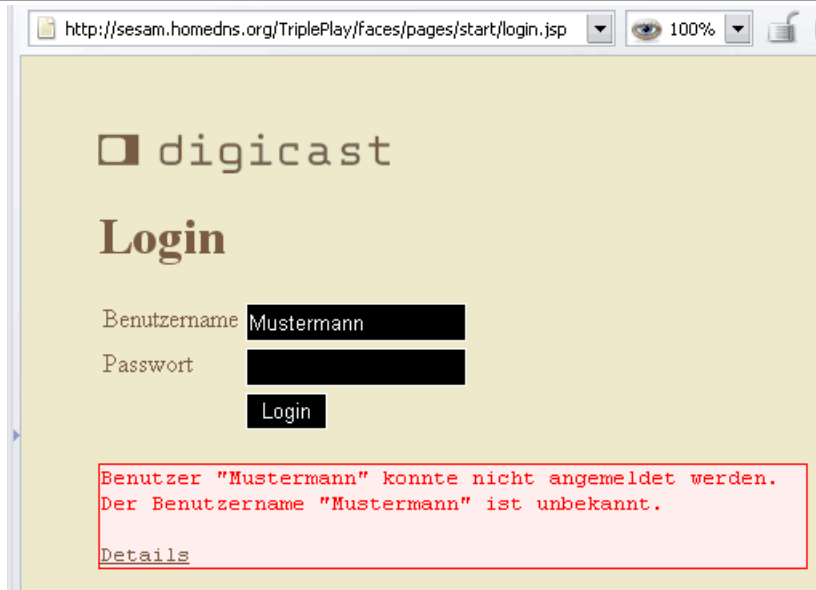
TriplePlay
05

Stunden: Schätzung 55,2, Verbraucht 38,3, Verbleibend 0,8

Story	!	Task	Empfänger	Geschätzt	Verbraucht	Verbleibend	Aktionen
Portal	1	Clock		0,5	0,2	0,4	
Portal	1	Konferenz-Oberfläche für VoIP-Option verändern		8,0	8,0	0,0	
Portal	1	Refaktorisierung & Tests		4,8	4,8	0,0	
Portal	1	Spiegelung von Videotext Bildern+Links		4,0	3,6	0,4	
Portal	1	Der Telefon-Server soll für VoIP Unterstützung konfiguriert werden (SipGate)		5,0	0,3	0,0	
Portal	1	Der Telefon-Server soll mit einer zweiten ISDN-Leitung arbeiten können.		10,0	1,0	0,0	
Portal	1	login logging		0,5	0,3	0,0	
Portal	1	Multiple User support		2,4	2,4	0,0	
Portal	1	Spring Refaktorisierung		20,0	17,8	0,0	

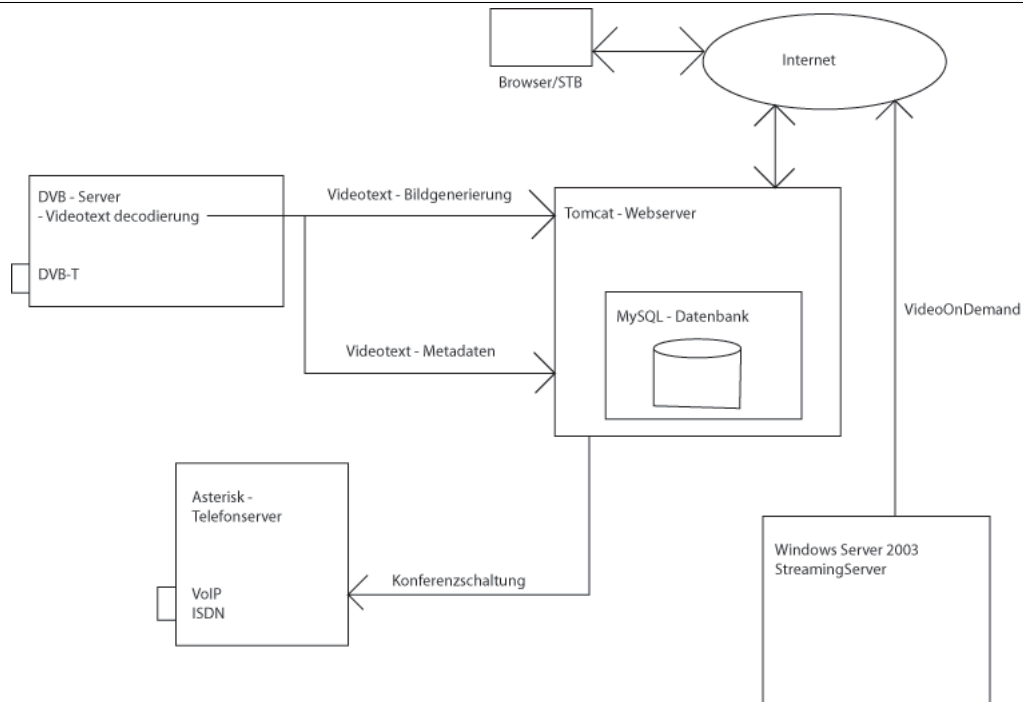
Abgeschlossen Nicht zugewiesen Nicht geschätzt

Abbildung 2.6 Login



2.4 Serverstruktur

Abbildung 2.7 Beispiel Server Struktur



Entsprechend der **Abbildung 2.7** wurden beim Kunden einige Server für das Projekt installiert und konfiguriert. Dazu zählen:

- DVB-Server zum Empfang der DVB-Daten², Generierung der Videotextbilder und Speicherung der Videotext-Metadaten (Verweise auf andere Seiten) in einer Datenbank.
- Streaming Server (Windows Server 2003) zur Bereitstellung von Video und Audiomaterial.
- J2EE Application Server (Tomcat) für das Web-Projekt
- Datenbank-Server (MySQL) zur Speicherung der Videotext-Metadaten
- Telefon-Server (Asterisk) für die Konferenzschaltungsfunktion
- STB als Endgeräte zur Darstellung der HTML-Seiten

2.5 UML

Es folgen einige UML-Diagramme, welche die Software-Architektur genauer visualisieren.

²Es wurde eine Hauppauge Nova-T PCI-Karte verwendet

Abbildung 2.8 Übersicht der Paketabhängigkeiten

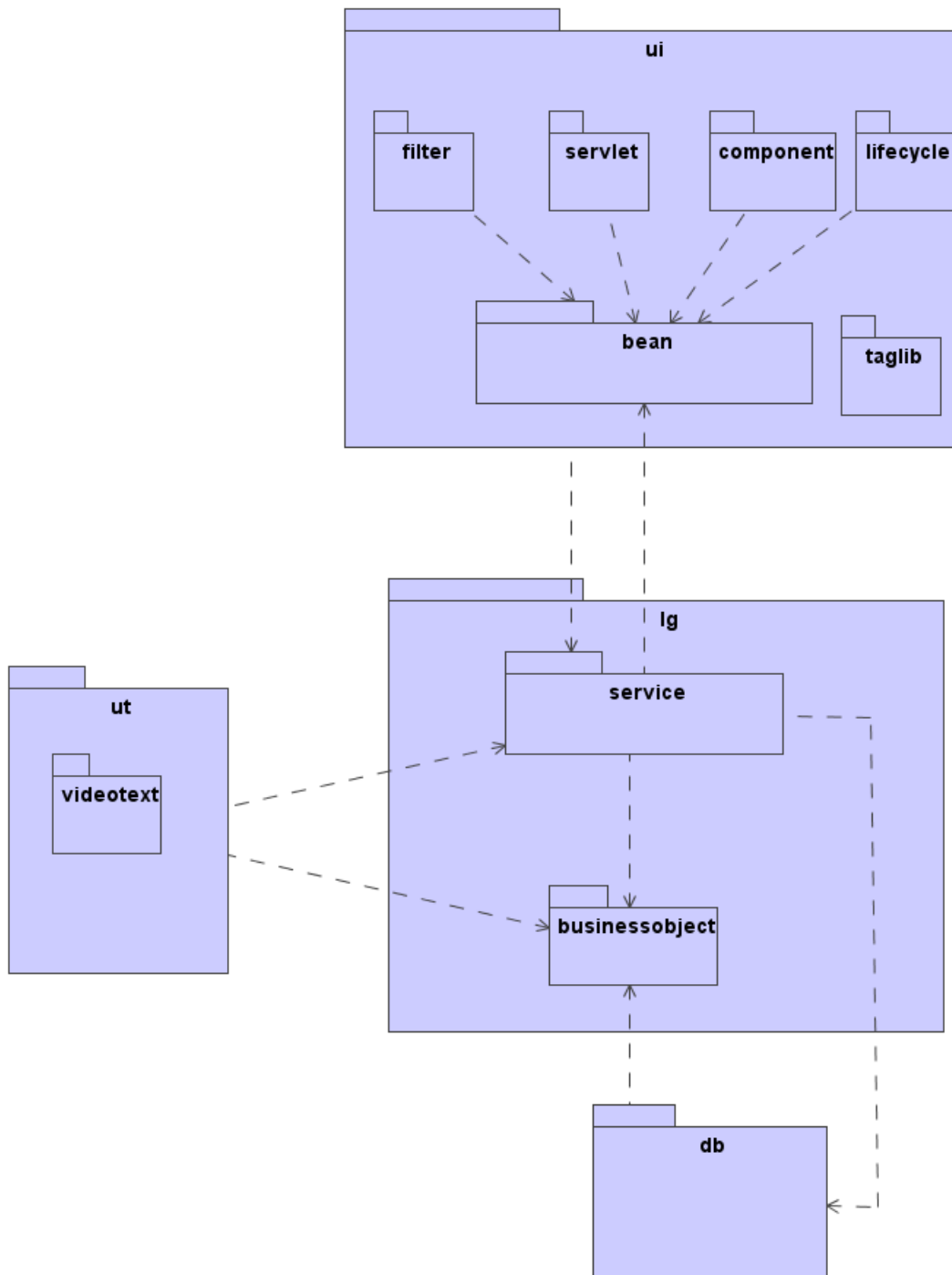


Abbildung 2.9 Fachklassendiagramm

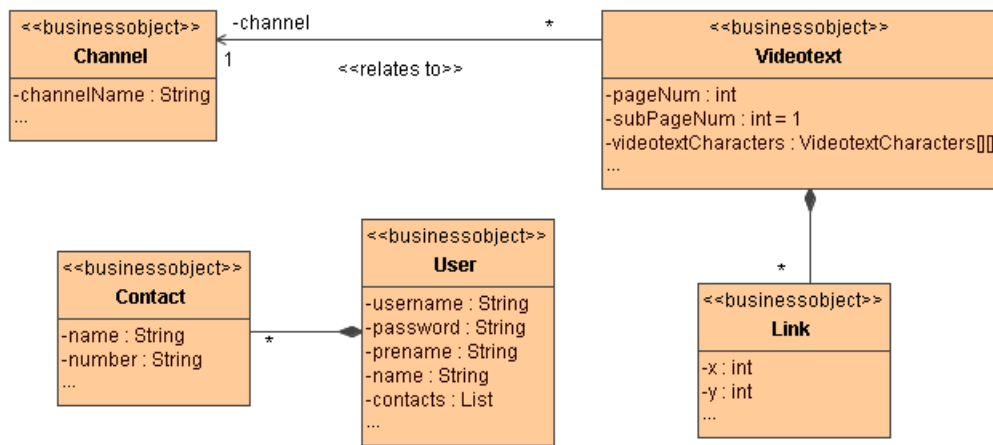


Abbildung 2.10 Übersicht Dao-Schnittstellen

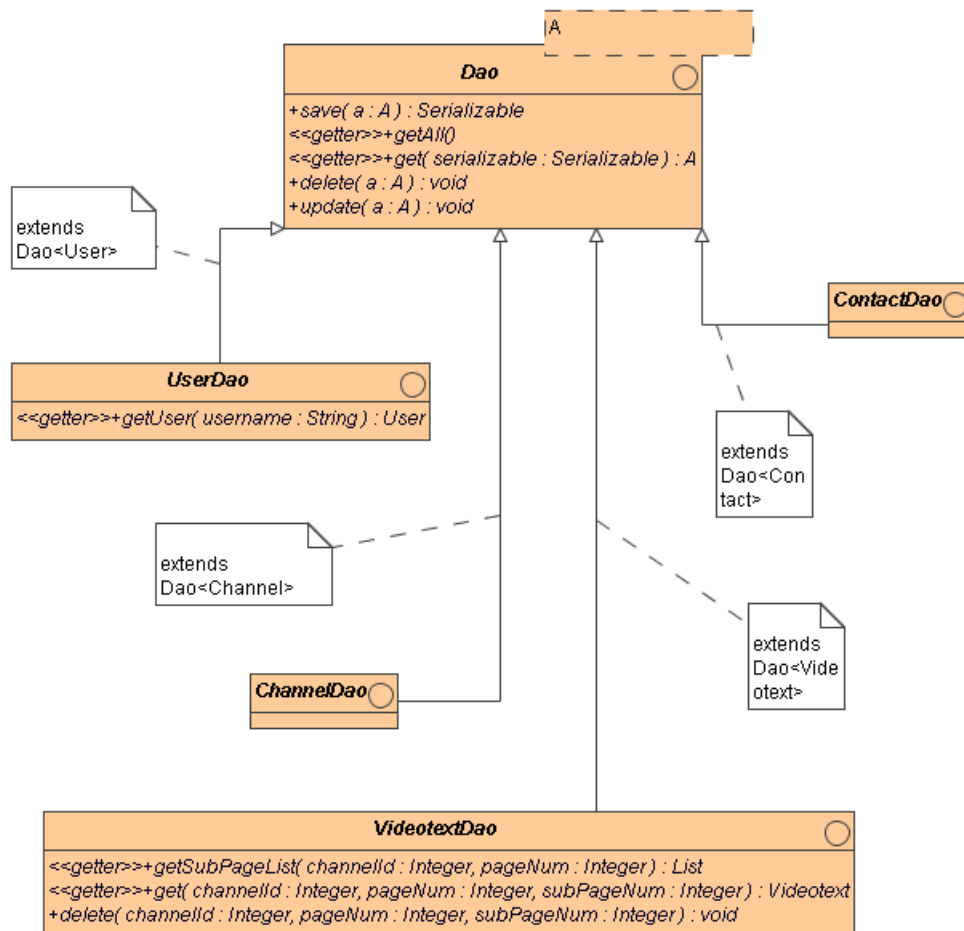
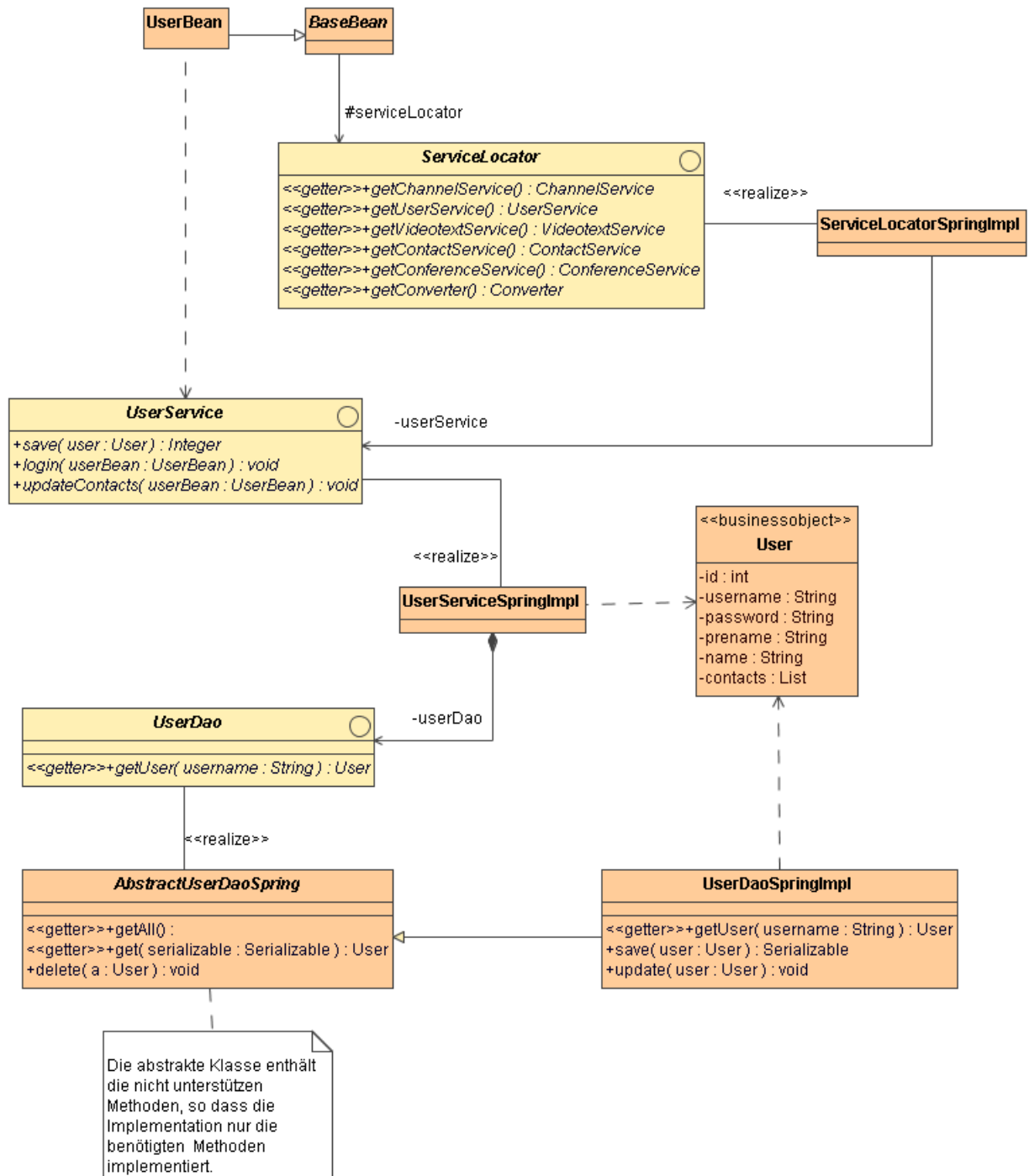


Abbildung 2.11 Übersicht von der UI zur DB



Kapitel 3

Frameworks

3.1 Spring Framework

3.1.1 Das Spring Framework

Spring ist ein Open Source Applikationsframework mit der Intention, J2EE Entwicklung zu vereinfachen und gute Programmierpraktiken zu fördern [Johnson]. Es besteht aus einem IoC-Container, einem Framework für die Konfiguration von Komponenten und einer Menge von modularen Diensten für Transaktionen, Persistenz und Web-Schnittstellen [Jose].

3.1.2 Konfiguration

Die **Abbildung 3.1** stellt eine Visualisierung der Konfigurationsdatei ¹ für das Spring Framework dar. Sie wurde mit dem Eclipse Plug-In „Spring IDE for Eclipse“ [SpringIDE] erstellt. In dieser Datei werden Klassen deklariert, die bei ihrer Instanziierung von anderen Klassen abhängig sind (In der Konfigurationsdatei werden die Klassen Beans genannt). Die Pfeile von einer Klasse zu anderen sind identisch zu den Assoziationspfeilen in UML.

Es folgt ein Auszug aus der Datei `applicationContext.xml`.

```
1 <beans>
2   ...
3   <bean id="DecoderContext"
4     class="com.digicast.ut.videotext.impl.DecoderContextImpl" />
5
6   <bean id="Decoder" class="com.digicast.ut.videotext.impl.DecoderImpl">
7     <property name="decoderContext">
8       <ref bean="DecoderContext" />
9     </property>
10  </bean>
11
12  <bean id="Converter" class="com.digicast.ut.videotext.impl.ConverterImpl">
13    <property name="decoder"><ref bean="Decoder" /></property>
14    <property name="fontFilename">
15      <value>com/digicast/lg/resource/videotext.ttf</value>
16    </property>
17    <property name="graphicFontSize"><value>17</value></property>
18    <property name="textFontSize"><value>15</value></property>
19  </bean>
20 </beans>
```

Hier werden die Klassen `DecoderContextImpl`, `DecoderImpl` und `ConverterImpl` konfiguriert. Die Klasse `ConverterImpl` benötigt beispielsweise eine Instanz der Klasse `DecoderImpl`, die wiederum eine Instanz der Klasse `DecoderContextImpl` verwendet. Die Übergabe der Instanzen erfolgt über Setter-Methoden in den jeweiligen Klassen.

¹Der Pfad zu dieser Datei lautet: `TriplePlay/WebContent/WEB-INF/applicationContext.xml`.

Des Weiteren können in dieser Konfigurationsdatei auch „primitivere“ Abhängigkeiten wie z.B. eine Zeichenkette, eine Zahl usw. konfiguriert werden. Als Beispiel dafür dient die Klasse `ConverterImpl`. Ihr wird die URL zur TrueType-Font zur Videotextgenerierung konfiguriert.

3.2 JavaServer Faces

JavaServer Faces [JSF] ist ein neues Framework von Sun, mit dessen Hilfe künftige Webanwendungen schneller und effizienter erstellt werden können. JSF ist dabei primär ein UI-Framework, das sich somit auf die Erstellung und Verarbeitung von Benutzerschnittstellen (User Interfaces) fokussiert hat.

3.2.1 Exception Framework

Alle von uns verwendeten Exceptions sind abgeleitet von der Basisklasse `Failure`, die wiederum von der Klasse `RuntimeException` abgeleitet ist. Um ständige try-catch-Blöcke zu vermeiden, haben wir Möglichkeiten von JavaServer Faces genutzt, eine eigene `Lifecycle`-Klasse statt der Standardimplementation anzumelden. In der `execute`-Methode der Klasse wird der Lebenszyklus einer JavaServer Faces Anfrage angestoßen. Werden bei der Ausführung Ausnahmen geworfen, so landen sie letztlich in der `execute`-Methode. Dort werden die zu der Klasse gehörenden internationalisierten Nachrichten² für den Benutzer dargestellt.

3.2.2 UI-Komponenten

JSF ist mehr als ein „reines“ Framework. Es wird zusätzlich ein kompletter Werkzeugkasten mitausgeliefert, mit dem Oberflächen nach dem Baukastenprinzip zusammengestellt und mit entsprechenden Validatoren und Eventmechanismen erstellt werden können.

Es können jedoch auch eigene Komponenten erstellt werden. Dies wurde für das Projekt an drei Stellen verwendet:

Die Klasse `UIVideotextSubPageLinks`³ ist eine Komponente, die eventuell vorhandene Unterseiten einer gerade angezeigten Videotextseite auflistet und verlinkt (siehe [Abbildung 2.2](#)). Im Paket `com.digicast.ui.taglib` ist ein entsprechender JSF-Tag definiert, der in `videotext.jsp` benutzt wird. Eine ausführlicheres Beispiel ist unter [\[JSFComponents\]](#) zu finden.

Weitere selbst entwickelte JSF-Komponenten sind eine Fehlerausgabe mit Verlinkung zu einer Detailseite⁴ und eine Komponente für die HTML ImageMap, welche direkte Verlinkungen auf einer Videotextseite ermöglicht. Zu finden sind die Klassen im Paket `com.digicast.ui.component`.

Es folgt ein Auszug aus der Datei `pages/start/start.jsp` in der die Verwendung einer eigenen Komponente mit zugehörigem Namensraum gezeigt ist.

```

1 <%@ taglib uri="http://java.sun.com/jsf/html" prefix="h"%>
2 <%@ taglib uri="http://java.sun.com/jsf/core" prefix="f"%>
3 <%@ taglib uri="http://www.digicast.com" prefix="d"%>
4 ...
5
6 <html>
7 ...
8     <d:errorMessage styleClass="errorMessage"/>
9 ...
10 </html>
```

3.3 Hibernate

3.3.1 Hibernate - Relational Persistence For Idiomatic Java

Hibernate ist ein Open Source OR-Mapping Tool, welches sich in den letzten Jahren in der Entwicklergemeinschaft neben den großen Standards JDO [JDO] und EJB [EJB] als interessante Alternative bewährt hat.

²`com/digicast/ui/resource/Messages.properties`

³im Paket `com.digicast.ui.component`

⁴`pages/details.jsp`

Hibernate bietet eine Abstraktionsschicht gegenüber der darunterliegenden Datenbank und ermöglicht es so, die Datenbank schnell zu wechseln.⁵ Dabei versucht das Projekt, Konzepte des objektorientierten Designs auf eine relationale Datenbank abzubilden.⁶

3.3.2 Datenbankschema

Abbildung 3.2 ist das aktuelle Datenbankschema. Die Synchronität von Schema und Quellcode wird dadurch gewährleistet, dass es aus speziellen xDoclet-Tags innerhalb der Java Sourcecode-Dateien generiert wird. Eine Detaillierte Beschreibung ist in Siehe [Abschnitt 3.3.3](#) zu finden.

3.3.3 xDoclet - Generierung

Es folgt ein Auszug aus der Klasse `Link` in der Datei `Link.java`.

```
1 package com.digicast.lg.businessobject;
2
3 /**
4  * @hibernate.class
5  */
6 public class Link {
7     private int id;
8     private int x;
9     private int y;
10    private String link;
11
12    /**
13     * standard constructor
14     */
15    public Link() {}
16
17    /**
18     * @param x the column of the link
19     * @param y the row of the link
20     * @param link the value of the link
21     */
22    public Link(final int x, final int y, final String link) {
23        this.x = x;
24        this.y = y;
25        this.link = link;
26    }
27
28    /**
29     * @return the id (generated by Hibernate)
30     * @hibernate.id generator-class="native" unsaved-value="0"
31     */
32    public int getId() {
33        return id;
34    }
35
36    /**
37     * @return the link value
38     * @hibernate.property
39     */
40    public String getLink() {
41        return link;
42    }
```

⁵Zur Zeit unterstützt Hibernate 20 sogenannte SQL-Dialekte. Unter anderem MySQL, Oracle 9/10g, Microsoft SQL Server, Progress und Pointbase.

⁶Dazu gehören u.A. Assoziationen, Vererbung, Polymorphie und Komposition.

```
43
44     /**
45      * @return the column number
46      * @hibernate.property
47      */
48     public int getX() {
49         return x;
50     }
51
52     /**
53      * @return the row
54      * @hibernate.property
55      */
56     public int getY() {
57         return y;
58     }
59     ...
61 }
62 }
```

Mit speziellen xDoclet anweisungen beschreibt man die Art und Weise der Abbildung der Klassen auf die Tabellen in der Datenbank. Diese Anweisungen werden durch einen Speziellen Ant-Task⁷ in die von Hibernate verwendeten *.hbm.xml Dateien transformiert.

Durch diese Vorgehensweise ist es möglich, die Abbildungsdateien völlig synchron zum Quelltext zu halten.

3.3.4 SchemaExport

Wie in [Abschnitt 3.3.3](#) beschrieben, ist es einfach, die Abbildungsdateien synchron zum Quelltext zu halten.

Über diesen Aspekt hinaus gibt es die Möglichkeit, durch einen weiteren Ant-Task⁸ aus den Abbildungsdateien eine SQL-Skriptdatei zu generieren, die die benötigten SQL-Anweisungen [\[SQLTutorial\]](#) für das Erstellen der Tabellen in einer Datenbank enthält.

3.3.5 Fazit

Durch die in [Abschnitt 3.3.3](#) und in [Abschnitt 3.3.4](#) beschriebenen Vorgehensweisen ist bei der Entwicklung keine Zeit für das Datenbankschema benötigt worden. Lediglich die gegenseitigen Assoziationen der Klassen waren bei der Planung wichtig. Dies aber wird bereits von UML-Klassendiagrammen abgedeckt, womit der Einsatz von ER-Diagrammen überflüssig war.

⁷Zu finden sind die Ant-Tasks in der Datei `build.xml`. Der Entsprechende Task heißt `hbm`.

⁸Der Entsprechende Task heißt `schemaexport`

Abbildung 3.1 Visualisierung der Spring Konfigurationsdatei

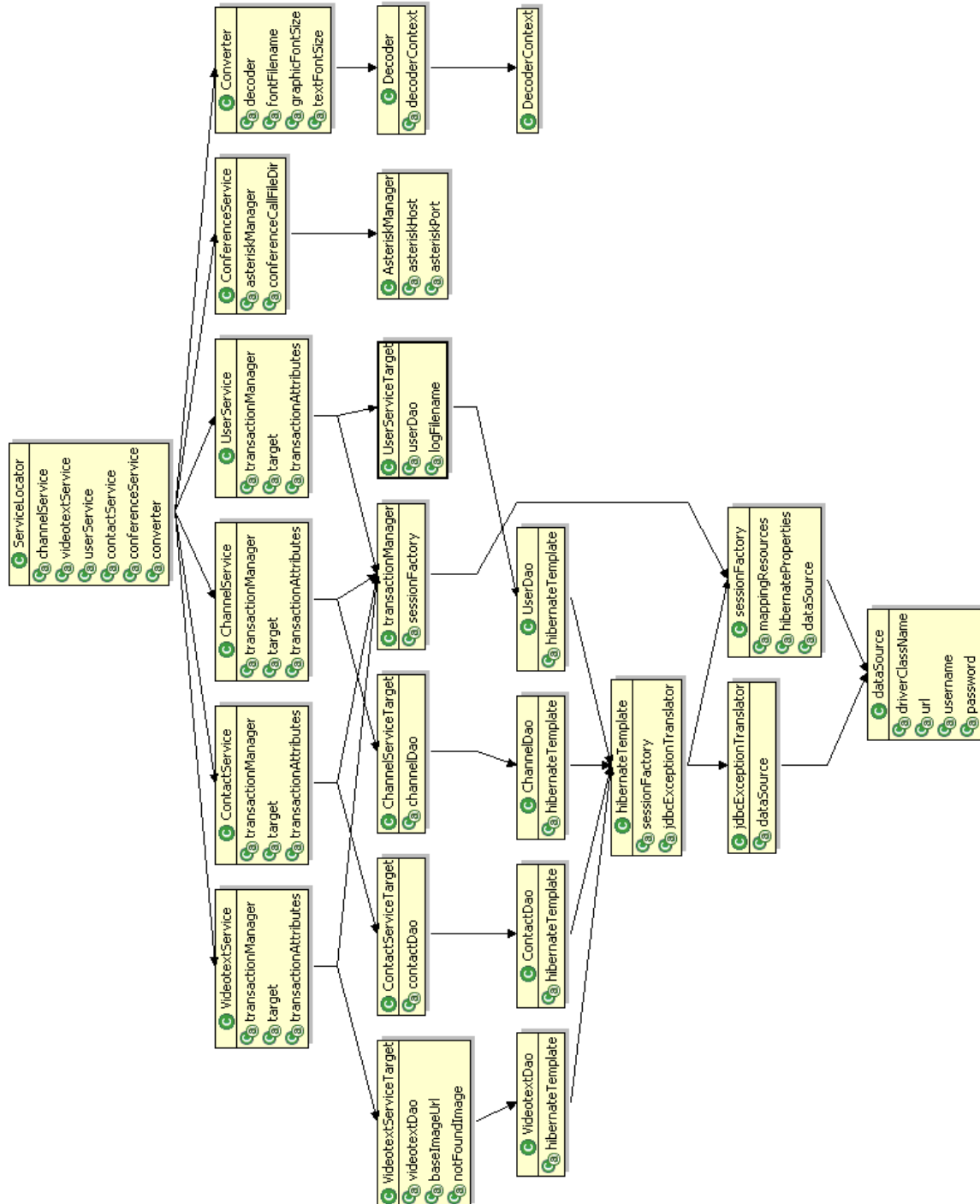
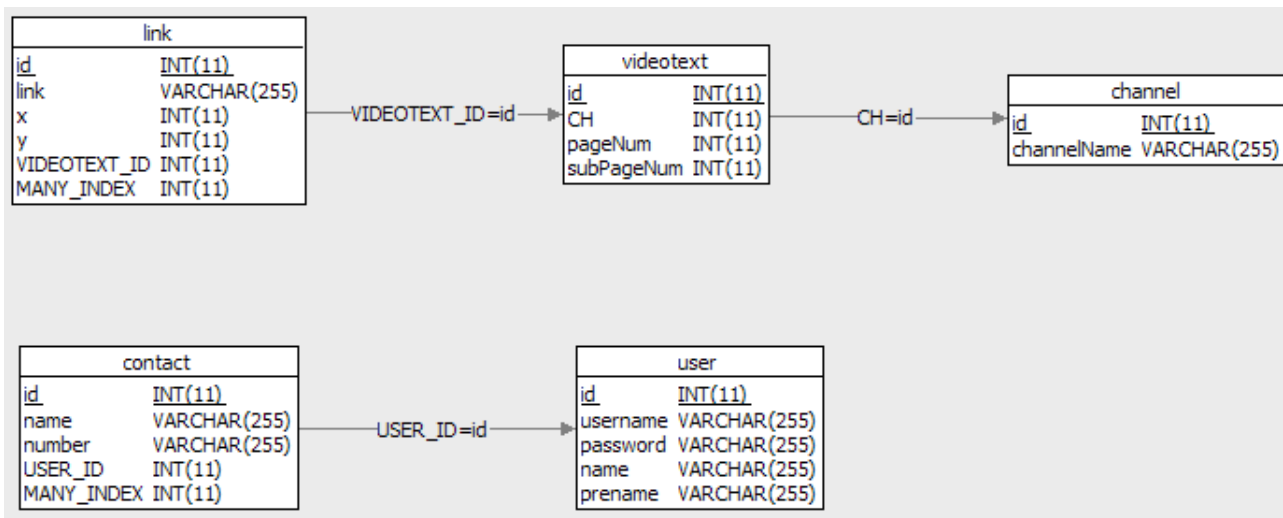


Abbildung 3.2 Das Datenbankschema



Kapitel 4

Oberflächentestprotokoll

4.1 login.jsp

4.1.1 Benutzername

Beim Aufruf mit einem nicht vorhandenen Benutzernamen erscheint die folgende Fehlermeldung (Beispiel):

```
Benutzername: 123
Passwort     : xyz

"Benutzer "123" konnte nicht angemeldet werden.
Der Benutzername "123" ist unbekannt.

Details
```

Beim Betätigen des Login-Links ohne Angabe von Benutzername und Passwort erscheinen zwei Meldungen, dass die jeweiligen Werte benötigt werden.

4.1.2 Passwort

Beim Aufruf mit einem vorhandenen Benutzernamen und einem falschem Passwort erscheint die folgende Fehlermeldung (Beispiel):

```
Benutzername: 1
Passwort     : 123

Benutzer "1" konnte nicht angemeldet werden.
Das angegebene Passwort ist falsch.

Details
```

4.1.3 Details

Betätigen des Links mit der Beschriftung *details* bei einer Fehlermeldung führt einem zur `details.jsp` Seite.

4.2 details.jsp

4.2.1 Benutzername

Betätigen des Links mit der Beschriftung *Zurück* führt einem zu `login.jsp`.

4.3 start/start.jsp

4.3.1 Benutzername

Es erscheint eine Begrüßung. Diese sollte übereinstimmen mit dem Vornamen und dem Nachnamen des angemeldeten Benutzers (Beispiel):

```
Willkommen Jérôme Richter!
```

4.3.1.1 Logout

Nach Betätigung des Logout-Links gelangen sie zurpck zur Login-Seite.

4.4 phone/contacts.jsp

Beim Betätigen des Hinzufügen-Links ohne Angabe von Name und Nummer erscheinen zwei Meldungen, dass die jeweiligen Werte benötigt werden.

Nach dem Hinzufügen von Name und Nummer erscheint der Eintrag am Ende der Liste.

Nach dem Betätigen des Löschen-Links eines Eintrags wird dieser aus der Liste entfernt.

Wurde ein neuer Eintrag hinzugefügt und der Abbrechen-Link gedrückt, ist dieser nach nochmaligem Aufruf der Seite nicht mehr in der Liste enthalten. Wurde jedoch der Speichern-Link gedrückt, so ist der Eintrag auch nach nochmaligem Aufruf der Seite vorhanden.

Beim erstellen eines neuen Kontakts mit der Nummer ÄBCünd anschließendem Speichern erscheint eine Fehlermeldung, dass die Nummer ungültig sei.

Kapitel 5

Programmierrichtlinien

5.1 Allgemein

Für das Projekt *TriplePlay* gelten die Programmierrichtlinien der Firma *digicast*. Die Kernpunkte, die für die gewählte Programmiersprache Java relevant sind, sowie deren eventuelle Anpassung auf das Projekt, werden im Folgenden beschrieben und sind verbindlich.

5.2 Sprache

Alle Pakete, Klassen, Methoden, Variablen, Annotationen, etc. sowie Kommentare müssen in Englisch verfasst sein.

5.3 Kommentare

Jede Datei muss einen Kommentar in folgender Form nach der Paketdefinition enthalten:

```
/**
 * <p>
 * project: TriplePlay<br/>
 * filename: filename.java<br/>
 * created on: 09.12.2004 18:33:45
 * </p>
 */
```

Durchgehende *javadoc*-Kommentare für die API-Klassen sind erwünscht.

5.4 Deklarationen

Es wird sich an die allgemeine Java-Konvention zur Benennung gehalten.

5.4.1 Konstanten, Variablen, Parameter

Konstantennamen sind groß zu schreiben.

```
public static final String CLASS_FILE_SUFFIX = ".class";
```

Variablenamen beginnen mit einem Kleinbuchstaben.

```
int amountOfCards = 4;
```

Parameter werden bezüglich ihrer Nutzung als in/out/inout Parameter mit dem entsprechenden Präfix „i-“, „o-“ oder „io-“ gekennzeichnet.

```
public void setName(final String i_name);
```

5.4.2 Methoden

Methodennamen beginnen mit einem Kleinbuchstaben.

```
public void setName(final String i_name);
```

5.4.3 Pakete

Pakete werden kleingeschrieben. Folgende Namenskonvention muss eingehalten werden:

- *ui* - Für das Paket User Interface
- *lg* - Für das Paket Logic
- *db* - Für das Paket Database

5.5 Einrückungen, Whitepaces und Klammerung

Einrückungen sollen mit Tabulatoren und nicht mit Leerzeichen realisiert werden. Whitespaces am Ende einer Zeile sollen nach Möglichkeit entfernt werden. Nach *for*, *while*, *if*, etc. folgt möglichst kein Leerzeichen. Öffnende Klammern werden auf die gleiche Zeile gesetzt, wie das Schlüsselwort, das einen neuen Block einleitet.

```
class A {
    public void B() {
        switch(condition) {
            ...
        }
    }
}
```

Folgt auf eine Bedingung nur eine einzige Anweisung, so ist auch diese in geschweifte Klammern zu fassen.

```
while(true) {
    doSomething();
}

if(numbers < 5) {
    numbers = 5;
}
```

5.6 TODOs

TODOs werden komplett groß geschrieben, damit diese in Umgebungen wie Eclipse geparkt werden können.

```
//TODO: write some stuff here...
```

Gleiches gilt für

- *HACK* - für Codebereiche, die gezielt nochmals refaktoriert werden sollten
- *FIXME* - für die Markierung gefundener Bugs

Kapitel 6

Fazit

6.1 Das Projekt

Dieses Softwareprojekt im Rahmen der Lehrveranstaltung SWP1 und SWP2 bei Prof. Knabe hat zwar einiges an Zeit in Anspruch genommen, jedoch haben wir auch ein breites Spektrum an neuen innovativen Techniken kennengelernt, welche uns vorher entweder unbekannt waren oder nur in relativ kleinen Umfang genutzt wurden.

Es handelte sich um ein ziemlich spannendes Projekt, sowohl inhaltlich als auch technisch. Die Implementierung eines Videotext-Dekoders war ebenso eine interessante Herausforderung, wie die Anpassung an einen technisch relativ beschränkten Web-Browser einer Set-Top Box.

Ein Web-Projekt erweist sich schließlich als tückisch, da (in unserem Falle zumindest) viele verschiedene Komponenten und Techniken zusammenspielen müssen (JSP/Servlet Container, JavaServer Faces, Hibernate). Allein die Umgebung zu installieren und zu konfigurieren ist aufwendig und war an der TFH nur unter erschwerten Bedingungen möglich. Ein Problem war der mangelnde Speicherplatz, der uns nur zögerlich gewährt wurde. Ein weiteres war, dass wir mit Java 1.5 (respektive 5.0) auf der Höhe der Zeit sein und die Vorteile der neuen Spracherweiterungen nutzen wollten. Da (zumindest im Wintersemester 2004/05) jedoch keine Entwicklungsumgebung im Softwarelabor eingerichtet war, wurden wir wirvor zusätzliche Probleme gestellt. Dies führte dazu, dass wir an der Fachhochschule eher weniger programmierten.

TriplePlay wurde am Ende eher für Demo-Zwecke und Proof-Of-Concepts gegenüber potentiellen neuen Kunden der Firma *digicast* verwendet. Als uns bewusst wurde, dass der Umfang des Projektes nicht weiter zunehmen würde, standen vorallem die eingesetzten Techniken gegenüber dem Inhalt des Programms im Vordergrund.

6.2 Erfahrungen mit Extreme Programming

Extreme Programming half uns darin, die optimalen Techniken für das Projekt zu finden und auch flexibel genug zu sein, neue Wege einschlagen zu können und alte Ideen relativ mühelos verwerfen zu können. An das testgetriebene Entwickeln musste man sich zunächst einmal gewöhnen. Es handelte sich um eine für uns völlig neue Herangehensweise, die sich sehr von den Entwurfstechniken unterscheidet, die wir bis dahin gelernt und angewand hatten. Dies konsequent durchzuhalten fiel uns zunächst etwas schwer, doch wir haben den Vorteil erkannt und halten es für eine äusserst interessante Alternative zu anderen gängigen Praktiken.

Da wir die meiste Zeit gemeinsam in der Firma gearbeitet haben, hatten wir immer direkten Kontakt zum Kunden, um neue Wünsche gleich berücksichtigen und evtl. Rückfragen stellen zu können. Der Auftraggeber hat auch großen Wert darauf gelegt, immer auf dem Laufenden zu sein und über Schwierigkeiten informiert werden zu können. So hatte dieser die Möglichkeit, flexibel den Projektverlauf beeinflussen zu können. Dies führte dann schließlich auch dazu, dass wir das ursprüngliche Szenario ([Abschnitt 1.2](#)) weitestgehend auf die Generierung und Darstellung von Videotext-Seiten beschränkt haben. Desweiteren ist auch die Integration des in der Firma vorhandenen Telefonservers eine Idee gewesen, die während der Projektrealisierung entstand.

Für die Realisierung notwendige technische Hilfsmittel wurden uns von der Firma zur Verfügung gestellt. Auch durften wir die STB nach Hause mitnehmen, um dort Browser-Tests durchführen zu können.

Die für XP typischen 2-wöchigen Iterationsschritte wurden ziemlich konsequent eingehalten und führten zu schnellen Ergebnissen. Die Zufriedenheit des Kunden bestätigte uns darin, dass wir für dieses Projekt den richtigen Weg wählten.

Wir hatten Glück, mit *digicast* einen Kunden zu haben, der sich sofort auf Extreme Programming einlies. Wir würden uns persönlich wünschen, dass eine solche Vorgehensweise mehr Einzug in die Industrie findet und Kunden sich gegenüber dem noch immer stark diskutierten Softwareentwicklungsprozess nicht mehr verschließen.

Persönlich fanden wir es ziemlich schade, dass wir aufgrund unvorhersehbarer Ereignisse und unglücklichen Konstellationen nur ein 2er-Team waren. Dadurch können XP-Praktiken wie Pair-Programming mit ständig wechselnden Mitarbeitern nicht wie erhofft durchgeführt werden. Man hat sich zwar auch so viel ausgetauscht, jedoch hätten weitere kreative Programmierer dem Projekt vielleicht zu ganz anderen interessanten Ideen verholfen.

6.3 Beurteilung der Lehrveranstaltung

Was für uns als XP-Gruppe besonders herausforderte, war die Tatsache, dass wir in mancher Hinsicht der Vorlesung voraus waren und uns eigenständig Techniken wie JUnit-Tests und zentrales Exception Framework erarbeiten mussten, bevor diese in der Lehrveranstaltung besprochen wurden. Dies lag jedoch in der Natur des Entwicklungsmodells für das wir uns bewusst entschieden hatten und lies sich demnach nicht vermeiden. Es wurde zwar auch auf gängige XP-Methoden eingegangen, jedoch lag der Schwerpunkt eher bei dem klassischen Analysemodell, was ja auch von dem Großteil der Gruppen durchgeführt wurde.

Die sporadischen Präsentationen des „status quo“ der Projekte waren interessant und manchmal haben sich auch durchaus hilfreiche Diskussionen über Problemlösungen und -ansätze ergeben. Für uns persönlich war es jedoch problematisch, dass wir relativ früh einen bereits funktionierenden Prototypen zeigen konnten, während andere ausführlich ihre Analysedokumente vorstellten. Die schnellen Ergebnisse lagen in der Natur des Extreme Programmings, was jedoch zur Folge hatte, dass sich für spätere Präsentationen (nachdem wir auch über unsere verwendeten Techniken berichtet hatten) wenig Neues zur Darbietung bot.

Die Wöchentlichen Rücksprachen zeigten fast immer neue Stellen auf, an denen noch Verbesserungen getätigt werden konnten. Gut war zudem, dass Prof. Knabe unseren fortgeschrittenen Projektstatus erkannte und daraufhin durchaus auch mal flexibel mit den Terminen sein konnte. Profitiert haben wir vor allem durch die gemeinsame direkte Analyse unseres Codes, wobei dies leider eher selten durchgeführt wurde.

Anhang A

Verwendete Programme

Folgende Programme wurden bei der Entwicklung des Softwareprojekts eingesetzt:

BETRIEBSSYSTEME

- Microsoft Windows XP
<<http://www.microsoft.com/windowsxp/default.mspx>>
- Debian Linux 3.0
<<http://www.debian.org>>

ENTWICKLUNGSUMGEBUNG

- Eclipse
<<http://www.eclipse.org>>
- Eclipse Plugins
 - Lombok
<<http://forge.objectweb.org/projects/lombok>>
 - Sysdeo Eclipse Tomcat Launcher
<<http://www.sysdeo.com/eclipse/tomcatPlugin.html>>
 - Faces Console
<<http://www.jamesholmes.com/JavaServerFaces/console/>>
 - Spring IDE
<<http://www.springframework.org/spring-ide/eclipse/>>
 - Log4eclipse
<<http://sourceforge.net/projects/log4eclipse/>>
 - Log4E
<<http://log4e.jayefem.de/index.php/Main.Page>>
 - Azuri clay - Database Modeling in Eclipse
<<http://www.azzurri.jp/en/software/clay/index.jsp>>

VERSIONSKONTROLLE

- CVS
<<http://www.cvshome.org>>

EXTREME PROGRAMMING

- XPlanner
<<http://www.xplanner.org>>

SERVER

- Microsoft Windows Server 2003
<<http://www.microsoft.com/windowsserver2003/default.msp>>
- MySQL
<<http://www.mysql.de>>
- Apache Tomcat
<<http://jakarta.apache.org/tomcat/>>

PROGRAMMIERSPRACHEN

- J2SE 5.0
<<http://java.sun.com/j2se/1.5.0/>>
- JSP
<<http://java.sun.com/products/jsp/>>

TOOLS / FRAMEWORKS

- JavaServer Faces
<<http://java.sun.com/j2ee/javaserverfaces/index.jsp>>
- Spring Framework
<<http://www.springframework.org>>
- Hibernate
<<http://www.hibernate.org>>
- Ant
<<http://ant.apache.org>>
- JUnit
<<http://www.junit.org>>
- xDoclet
<<http://xdoclet.sourceforge.net/xdoclet/index.html>>
- Log4j
<<http://logging.apache.org/log4j/docs/>>
- Ethereal
<<http://www.ethereal.com>>
- DVBtext
<<http://www.linuxstb.org/dvbtext/index.shtml>>
- LinuxTV-DVB-Treiber
<<http://www.linuxtv.org>>
- LinuxTV-DVB-Apps
<<http://www.linuxtv.org>>

DOKUMENTATION / UML

- DocBook
<<http://www.docbook.org>>
<<http://docbook.sourceforge.net>>

- LaTeX
<<http://www.latex-project.org>>
- DB2LaTeX
<<http://db2latex.sourceforge.net>>
- Saxon XSLT Processor
<<http://saxon.sourceforge.net>>
- Together
<http://www.borland.com/products/downloads/download_together.html>
- Adobe Illustrator
<<http://www.adobe.com/products/illustrator/main.html>>

Literaturverzeichnis

- [EJB] *Enterprise Java Beans*, , <<http://java.sun.com/products/ejb/>> . 3.3.1
- [JDO] *Java Data Objects*, , <<http://java.sun.com/products/jdo/index.jsp>> . 3.3.1
- [Johnson] *Johnson, R.: Introducing the spring framework.*, , <<http://www.theserverside.com/articles/article.tss?l=SpringFramework>> . 3.1.1
- [Jose] *Jose, B.: The spring framework*, , <http://javaboutique.internet.com/tutorials/spring_frame> . 3.1.1
- [JSF] *JavaServer Faces*, Hans Bergsten, 0596005393, O'Reilly, <<http://www.oreilly.com/catalog/jsvrfaces/>> . 3.2
- [JSFComponents] *How To Create Your Own Java Server Faces Components*, , <<http://www.jsftutorials.net/components/index.html>> . 3.2.2
- [SpringIDE] *Spring IDE for Eclipse - Beans Configuration Support*, , <<http://www.springframework.org/spring-ide/eclipse/>> . 3.1.2
- [SQLTutorial] *SQL Tutorial*, , <<http://www.w3schools.com/sql/default.asp>> . 3.3.4
- [vtxCtrlChars] *Die Steuerzeichen des Videotext-Dekoders*, , <<http://www.tm.uka.de/de/~harbaum/atari/videotxt/tabelle.html>> . 1.3.1

Glossar

Bean Eine Java Klasse mit Getter- und Setter-Methoden.

DVB DVB bezeichnet in technischer Hinsicht die standardisierten Verfahren zur Übertragen von digitalen Inhalten (Fernsehen, Radio, Mehrkanalton, Raumklang, interaktive Dienste wie MHP, EPG und Teletext und weitere Zusatzdienste) durch digitale Technik. Durch Datenkompression (heute MPEG-2, in Zukunft evtl. auch MPEG-4 und andere Verfahren) können im Vergleich zur analogen Fernsehübertragung mehrere Programme auf einem Transponder in wählbarer Qualität übertragen werden. Je höher die Daten komprimiert werden, desto mehr Programme können zeitgleich auf einem Transponder übertragen werden. Durch die hohe Komprimierung kann allerdings die Bildqualität leiden.

MPEG2 MPEG-2 ist ein MPEG-Standard zur Videodekodierung mit Videokompression und Audiokodierung mit Audiokompression.

Multicast Multicast bezeichnet in einem Computernetzwerk eine Übertragungsart von einem Punkt zu einer Gruppe, auch Mehrpunktverbindung genannt. Multicast ist die übliche Bezeichnung für IP Multicast, das ein effizientes Senden von Daten an viele Empfänger zur gleichen Zeit in TCP-IP Netzwerken ermöglicht. Das passiert mit einer speziellen Multicast Adresse, ein IP Bereich der speziell dafür reserviert ist. Multicast wird auch meist im Zusammenhang mit Audio- und Videoübertragungen genannt, diese nutzen Protokolle wie RTP.

SIP Das Session Initiation Protocol (SIP) ist ein Netzwerkprotokoll zum Aufbau einer Kommunikationssitzung zwischen zwei und mehr Teilnehmern. Das Protokoll wird in den RFC 3261 (früher RFC 2543) spezifiziert. In der ist das SIP ein häufig angewandtes Protokoll.

STB Unter einer Set-Top-Box (englisch für Draufstellkasten, abgekürzt STB) versteht man in der ein Gerät, das an ein anderes angeschlossen um gemeinsam zusätzliche Funktionen anzubieten. Das Basisgerät ist meist ein und die STBen Geräte zur Bild- und/oder Ton-Generierung und/oder -Aufzeichnung.

Transport-Stream Der MPEG2 Transport Stream dient dazu, ein oder mehrere digitale Fernseh- oder Radioprogramme zu übertragen. Der MPEG2 Transport Stream wurde entwickelt, um Daten aller Art über verlustbehaftete Übertragungswege zu transportieren. Der Transport-Stream ist neu im MPEG2 Standard, da heißt er ist im MPEG1 Standard nicht vorhanden.

VoIP Voice over IP ist das Telefonieren über ein Computernetzwerk auf der Grundlage des Internet Protokolls. Wird die IP-Telefonie genutzt, um Gespräche über das Internet zu führen, spricht man von Internet-Telefonie.

WMV9 Windows Media Video(WMV) ist ein proprietärer Video-Codec von Microsoft und Teil der Windows Media-Plattform.

Abbildungsverzeichnis

1	Projektbeschreibung	
1.1	Aufbau einer Videotextseite	7
2	Realisierung	
2.1	Kanalauswahl	11
2.2	Videotext	12
2.3	Kontakte	13
2.4	Konferenzschaltung	14
2.5	Hauptmenü	15
2.6	Login	15
2.7	Beispiel Server Struktur	16
2.8	Übersicht der Paketabhängigkeiten	17
2.9	Fachklassendiagramm	18
2.10	Übersicht Dao-Schnittstellen	18
2.11	Übersicht von der UI zur DB	19
3	Frameworks	
3.1	Visualisierung der Spring Konfigurationsdatei	25
3.2	Das Datenbankschema	26

Tabellenverzeichnis