

**Fachhochschule Köln
Abteilung Gummersbach
Fachbereich Informatik
Studiengang Wirtschaftsinformatik**

Diplomarbeit

Thema:

**Konzeption und Umsetzung eines datenbankgestützten,
automatisierten Systems zur Erzeugung der
Programminformation eines Fernsehsenders zur Nutzung
im B2B-Angebot**

Eingereicht von
Patrick Urban
Matrikelnummer 110 148 29

1. Prüfer: Prof. Dr. Heide Faeskorn-Woyke
2. Prüfer: Prof. Dr. rer. nat., Dipl.-Inform. Kristian Fischer

Köln, den 14.12.2000

Ehrenwörtliche Erklärung:

Hiermit erkläre ich, daß ich diese Arbeit selbstständig verfaßt und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Köln, 14.12.2000

Patrick Urban

Inhaltsverzeichnis

1	VORWORT	6
1.1	Einleitung	6
1.2	RTL Television	7
2	AUFGABENSTELLUNG	8
2.1	Problembeschreibung.....	8
2.2	Aufgabenstellung.....	8
3	GRUNDLAGEN	10
3.1	Unternehmensinterne Grundlagen	10
3.1.1	Programmwoche	10
3.1.2	Bildwoche	11
3.1.3	PROFIS	11
3.1.4	TRAFFIC	11
3.2	Allgemeine Grundlagen.....	12
3.2.1	Datenbank	12
3.2.2	ASP	26
4	IST-ANALYSE	35
4.1	Programmwoche.....	35
4.2	Bildwoche	36
4.3	Schnittstellenbeschreibung zu PROFIS und TRAFFIC	37
4.3.1	Planung des Sendeablaufs	37
4.3.2	Vorhandene Daten für die Programmwoche	38
4.3.3	Datenbankbeschreibung	41
5	SOLL-ANALYSE	46
5.1	Anforderungen der neuen Programmwoche	46
5.1.1	Programmwoche	46
5.1.2	Änderungen	46
5.1.3	Personen	47
5.1.4	Bildwoche	49
5.2	Geplanter Vorgang zur Erstellung der Programmwoche.....	50
5.2.1	Wegfindung	50
5.2.2	Beschreibung des Ablaufs zur Erstellung der neuen Programmwoche	53
6	BENUTZTE WERKZEUGE.....	55
6.1	DataArchitect	55
6.1.1	Einführung	55
6.1.2	Anwendungsbeispiel	55

6.2	Dreamweaver UltraDev.....	60
6.2.1	Einführung	60
6.2.2	Anwendungsbeispiel	61
7	DATENMODELL	65
7.1	Datenbankmodell der Programmwochen	65
7.2	Programmwoche.....	68
7.2.1	Programmwoche	68
7.2.2	Label	71
7.2.3	Bestandsobjekt	73
7.2.4	Material	78
7.2.5	Version	81
7.3	Änderungen	81
7.3.1	Elementare Änderungen	81
7.3.2	Redaktionelle Änderungen	83
7.3.3	Änderungsmeldungen	85
7.4	Personen	86
7.5	Sonstige.....	88
7.5.1	Wertebereich	88
7.5.2	Schlüsselverwaltung	89
7.5.3	Programmversionsverwaltung	89
7.6	Stored Procedures	90
7.6.1	GetAktuelleProgrammwoche	90
7.6.2	AenderungsuEberpruefung	92
7.6.3	VeroeffentlichePW	97
7.7	Views.....	99
8	PROTOTYP	101
8.1	Programmbeschreibung.....	101
8.2	Benutzerhandbuch	102
8.2.1	Programmwoche bearbeiten	102
8.2.2	Material bearbeiten	117
8.2.3	Person bearbeiten	121
9	ZUSAMMENFASSUNG.....	124
10	AUSBLICK	127

11	ANHANG	128
11.1	Abbildungsverzeichnis	128
11.2	Literaturverzeichnis und Quellenachweis	129
11.3	Abkürzungsverzeichnis	131
11.4	Glossar	132
11.5	Inhalt der CD-ROM.....	134
11.6	Erstellte Stored Procedures.....	135
11.7	Trigger auf den Tabellen.....	141

1 Vorwort

1.1 Einleitung

Ziel eines privaten Fernsehsenders ist es, möglichst hohe Einschaltquoten zu erreichen. Dazu ist es notwendig, die Zuschauer möglichst frühzeitig über das Programm und die Inhalte zu informieren. Ein wichtiges Medium zur Verbreitung dieser Information stellen Programmzeitschriften, Tageszeitungen usw. dar.

Zum Teil werden diese Zeitschriften ca. 2 Wochen vor dem Ausstrahlungstermin veröffentlicht. Vor der Drucklegung findet in den Verlagshäusern eine redaktionelle Aufbereitung (Layout, inhaltliche Bearbeitung, etc.) statt. Daher ist es notwendig, den Verlagen möglichst frühzeitig die Programminformationen zur Verfügung zu stellen.

Neben dem reinen Programmablauf liegt es im Interesse der Fernsehsender, auch mit ausführlichen Programmbesprechungen präsent zu sein. Aus diesem Grunde existiert im Hause RTL eine eigene Abteilung (Presse/Programminformation) für diese Aufgabe. Sie stellt sechs Wochen vor Ausstrahlung des Programms den kompletten Programmablauf einer Woche zur Verfügung. Dabei wird nicht nur der Programmplan weitergeleitet, sondern auch Inhaltsangaben zu Filmen und Serien, Informationen zu Darsteller usw. Zwischenzeitliche Änderungen im Sendeplan der davor liegenden Wochen werden in diesem Schritt ebenfalls an die Verlage weitergegeben. Zur Zeit werden alle Informationen in Word-Dokumenten zusammengefaßt. Eine Druckerei erstellt daraus ein Heft (die Programmwoche), welches den Verlagen zugestellt wird.

Diese Diplomarbeit hat zum Ziel, den zuvor beschriebenen Prozeß zu optimieren. Wesentlich hierfür ist eine Zeit- und Kostenersparnis durch Nutzung und Etablierung eines dem heutigen Stand der Informationstechnik entsprechenden Mediums. Dieses soll durch die Ablösung des Printmediums und den Aufbau eines Business-to-Business(B2B)-Angebotes im Internet erreicht werden.

Durch die Integration in die bei RTL vorhandenen Systeme soll eine Erhöhung der Informationsqualität und die Möglichkeit zur Wiederverwendbarkeit bereits erstellter Informationen erzielt werden.

Die Diplomarbeit untergliedert sich in folgende Kapitel:

Kapitel 2 beschreibt allgemein die Hintergründe und Ziele dieses Projektes.

Kapitel 3 erklärt Grundlagen, die zum Verständnis der gesamten Diplomarbeit notwendig sind.

Kapitel 4 gibt einen Einblick über den Arbeitsablauf bei der Erstellung der bisherigen Programm- und Bildwoche.

Kapitel 5 stellt den konzeptionellen Aufbau der neuen Programmwoche vor.

Kapitel 6 werden die Programme vorgestellt, die zur Realisierung des Projektes verwendet wurden.

Kapitel 7 erklärt das Datenmodell, daß der Programmwoche zu Grunde liegt.

Kapitel 8 stellt den Prototypen vor, der die Funktionalität der Programmwochendatenbank veranschaulicht.

Kapitel 9 und 10 gehen auf den bisherigen Stand und die zukünftigen Erweiterungen der Programmwochendatenbank ein.

1.2 RTL Television

RTL Television ist Deutschlands erfolgreichster TV-Sender. Das 1984 gegründete Unternehmen ist die Nummer 1 auf dem deutschen Zuschauermarkt - mit besonders großem Vorsprung bei den 14- bis 49-Jährigen. Der kontinuierliche Erfolg in dieser jungen Zielgruppe hat RTL auch zum Marktführer auf dem deutschen TV-Werbemarkt gemacht.

Bei RTL arbeiten ca. 1.800 Mitarbeiter. Von diesen sind nur 800 fest angestellt und der Rest arbeitet als freie Mitarbeiter in verschiedenen Projekten mit.

1999 konnte RTL seine Spitzenposition mit einem Werbeumsatz von 3,809 Milliarden Mark brutto und 2,434 Milliarden Mark netto noch weiter ausbauen. Zugleich erzielte RTL im vergangenen Jahr mit 368 Millionen Mark den höchsten Vorsteuer-Gewinn der Unternehmensgeschichte.

2 Aufgabenstellung

2.1 Problembeschreibung

Bisher werden alle Informationen zu geplanten Fernsehsendungen bei RTL von einer Redaktion aufbereitet und in mehreren Word-Dokumenten zusammengefaßt. Diese Programminformation erscheint wöchentlich und wird in gedruckter Form mit einer Auflage von ca. 300 Exemplaren auf herkömmlichem Weg versendet. Empfänger sind z.B. produzierende Redaktionen von Sendeformaten für RTL, Verlage von TV-Magazinen, Zeitungsverlage usw. Zugleich werden aus diesen Word-Dokumenten HTML-Seiten generiert und im Intranet veröffentlicht. Nicht alle Verlage von Fernsehzeitschriften nutzen dabei den direkten Kontakt mit RTL, sondern beschaffen sich die Informationen über Drittanbieter. Ein solcher Anbieter ist zum Beispiel die Deutsche Mailbox. Diese arbeitet die von RTL erhaltenen Informationen auf und stellt sie kleineren Zeitschriften zur Verfügung.

Der Arbeitsaufwand für die Erstellung der Dokumente und das Drucken der Programmwoche sind große Kostenfaktoren, die durch wiederverwendbare Daten und ein Online-Angebot gemindert werden könnten. Allein durch Wegfall der Printversion können im Jahr ca. 500.000 DM eingespart werden.

Mit Hilfe des Internetangebots kann RTL auch auf Drittanbieter verzichten und verkürzt damit die Kommunikationswege zum Endkunden. Durch den direkten Kontakt kann auf Wünsche und Vorschläge schneller eingegangen werden.

Deshalb soll Anfang 2001 die gedruckte Version der Programmwoche durch ein B2B-Angebot im Internet ersetzt werden.

2.2 Aufgabenstellung

Ziel der Diplomarbeit ist die Konzeption und Umsetzung eines datenbankgestützten, automatisierten Systems zur Erzeugung der Programminformation eines Fernsehsenders zur Nutzung im B2B-Angebot.

Dabei sollen die vorhandenen Daten aus der Bestandsverwaltung (PROFIS) und des Programmplanungssystems (TRAFFIC) für die Programmwoche genutzt werden und innerhalb einer eigenen Datenbank zusammengefaßt werden. Die Programmwochenpflege soll über ein FrontEnd innerhalb eines Internetbrowser realisiert werden. Bearbeitungen an der Programmwoche werden in der Datenbank gespeichert und sind immer wieder abrufbar.

Änderungen und Erweiterungen der redaktionellen Daten (Sendungsbeschreibungen, Personendaten) werden an die PROFIS-Datenbank zurückgegeben, so daß alle Bestandsdaten in allen Systemen stets einheitlich sind. Die Qualität der Bestandsdaten soll durch die

Bearbeitung über die Abteilung Presse/Programminformation zunehmen und wiederverwendbar sein.

Änderungen zwischen der Programmwochendatenbank und dem aktuell eingeplanten Stand von TRAFFIC sollen über einen Abgleich erkannt und automatisch übernommen werden. Zugleich soll für jede Änderung eine Meldung generiert werden, die später an die Verlage weitergegeben wird.

Ebenfalls wird eine Anbindung der Bildwochendatenbank angestrebt, um die Medien Bild und Text zusammenzuführen.

Diese Diplomarbeit umfaßt folgende Themen innerhalb dieses Projektes:

- Ist-Analyse der Strukturen und Abläufe innerhalb der alten Programmwoche
- Design der neuen Abläufe und Datenstrukturen
- Realisierung der Datenbank und der Businesslogik
- Machbarkeitsnachweis mit Hilfe eines Prototypen

Folgende Aufgaben sind **nicht** Teil dieser Diplomarbeit:

- Datenbankerweiterungen im Basissystem PROFIS
- Erstellen des endgültigen Clients zur Bearbeitung der Programmwoche
- Realisierung des B2B-Angebotes im Internet

3 Grundlagen

3.1 Unternehmensinterne Grundlagen

3.1.1 Programmwoche

Die Programmwoche ist ein ca. 100-seitiges Heft, welches wöchentlich herausgegeben wird. Es beinhaltet den kompletten Programmablauf mit einem Vorlauf von 6 Wochen. Neben dem Sendeplan steht zu jeder geplanten Sendung eine ausführliche Beschreibung. Abnehmer der Programmwoche sind unter anderem Verlage von TV-Magazinen, Zeitungsverlage und Redaktionen von Sendeformaten.

Die bisherige Programmwoche setzt sich aus folgenden Bestandteilen zusammen:

Titelblatt: Auf dem Titelblatt befindet sich die Nummer der aktuellen Programmwoche und der Gültigkeitszeitraum.

Highlights: In den Highlights werden ausgesuchte besondere Events bei RTL vorgestellt, wie z.B. die Topfilme und neue Serien dieser Programmwoche. Das Kapitel enthält ein Bild vom Topevent der Woche und Randinformationen zu jedem Highlight. Zugleich werden alle bekannten Darsteller vorgestellt, die innerhalb dieser Woche in Filmen oder Serien mitspielen.

Spielfilmvorschau: Alle Spielfilme der folgenden Programmwoche werden hier nach Datum sortiert aufgelistet. Der Leser erhält hier die Informationen über Label, Startzeit, Sendetitel, Genre, Produktionsland, Produktionsjahr, Länge, Originaltitel, mitwirkende Schauspieler, Regisseur und Datum der Erstausrahlung für jeden Spielfilm.

Kurzablauf: Der Kurzablaufplan ist eine Übersicht über alle Sendungen der aktuellen Programmwoche. Er enthält Informationen über die Label, Startzeit, Sendetitel und das Tonformat. Bei Serien, die nicht zum Kinderprogramm gehören, wird der Episodentitel in Deutsch angegeben. Sonderabläufe enthalten zusätzlich die Information über die Moderation. Die Filme beinhalten - wie in der Spielfilmvorschau - die Attribute Label, Startzeit, Sendetitel, Genre, Produktionsland, Produktionsjahr, Länge, Originaltitel, mitwirkende Schauspieler und Regisseur.

Langablauf: Der Langablauf ist der Hauptbestandteil der Programmwoche. Zusätzlich zu den Informationen aus dem Kurzablaufplan wird hier zu jeder Sendung das Genre, das Produktionsland und -jahr, der Originaltitel (auch bei einer Episode), der Erstausrahlungstermin, die FSK, ein Kurz- und ein Langinhalt, die Besetzungsliste und der Stab angegeben.

Änderungen: Die noch nicht publizierten Änderungen im Programmplan für die vorangegangenen 5 Programmwochen werden in diesem Kapitel zusammengefaßt. Zusätzlich zu den üblichen Attributen wie Sendedatum, Startzeit, Sendetitel usw. wird die Art der Änderungen und evtl. die Auswirkungen auf den Rest des Sendetages angegeben.

3.1.2 Bildwoche

RTL stellt den Verlagen wöchentlich Bilder zur aktuellen Programmwoche auf einem HTTP-Server zur Verfügung. Die Bilder können dort in zwei Qualitätsstufen (Thumbnail- und Layout-Qualität) mit einem Web-Browser angesehen werden. Diese können dann einzeln, als Bildauswahl oder komplett ausgewählt werden. Die Zusammenstellung wird in Repräsentation in einen Download-Verzeichnis für den User zusammen mit einer Informationsdatei abgelegt und kann per FTP heruntergeladen werden.

3.1.3 PROFIS

PROFIS ist ein **Programm-** und **Film**informationssystem, welches die Daten des Kerngeschäftes eines Fernsehsenders bereitstellt.

Die Anwendung besteht aus vier Komponenten, die zusammen mit TRAFFIC als Ganzes die Kernprozesse eines Fernsehsenders vom Filmeinkauf bis zum Werbezeitenvertrieb abdecken. Über Schnittstellen sind vorhandene Lösungen - wie z.B. Sendeablaufsteuerung oder SAP (Finanzbuchhaltung) - mit PROFIS zu einem Gesamtsystem integriert. Die einzelnen Komponenten von PROFIS sind:

- Lizenzrechteverwaltung
- Film- und Serienverwaltung
- Material- und Archivverwaltung
- Lizenzbuchhaltung

Zur Erstellung der Programmwoche wird der Datenbestand der Film- und Serienverwaltung benötigt. Alle Eigenschaften einer Sendung - wie Sendetitel, Genre, Produktionsland und Erstaussstrahlung - werden hier zentral verwaltet. Eine Übersicht der benötigten Daten für die Programmwochendatenbank befindet sich im Kapitel 4.3.2.

3.1.4 TRAFFIC

Traffic ist das Programmplanungssystem bei RTL. Es ist eine Eigenentwicklung, die über 6 Jahre hinweg den Bedürfnissen des Senders angepasst worden ist. Es unterstützt den gesamten operativen Prozeß der Programmplanung von der langfristigen Planung bis hin zur Erstellung

eines Ablaufes für das Sendeautomations-System. Über eine Anbindung an die Datenbank der Film- und Serienverwaltung von PROFIS ist es möglich, Sendungen auszuwählen und für die einzelnen Tage einzuplanen. So entsteht ein Scheduling-System für die Sendetage (Wann läuft welche Sendung).

3.2 Allgemeine Grundlagen

3.2.1 Datenbank

Alle Datenbankgrundlagen beziehen sich auf das bei RTL eingesetzte Datenbanksystem Adaptive Server Enterprise 12.0 von Sybase. Die Datenbank liegt auf einer HP9000 mit einer HP-UX Umgebung.

Der Adaptive Server Enterprise (ASE) ist eine Weiterentwicklung des SQL Servers. Bis zur Version 4.2 unterschieden sich die Produkte von Sybase und Microsoft nicht. Ab diesem Zeitpunkt gingen beide Firmen ihren eigenen Weg. Um Verwechslungen zu vermeiden, nannte Sybase ihr Datenbanksystem um und startete mit der Version 11.5.

Das aktuelle Release - der Adaptive Server Enterprise 12.0 - ist ein performantes, relationales Datenbankmanagementsystem für OLTP (On-Line Transaction Processing), operative Data Warehouses und Mixed Workload-Anwendungen. Es ist entwickelt worden, um die heutigen Bedürfnisse einer traditionellen und einer Internet-basierten Datenbank zusammen zu führen und wurde speziell in drei Schlüsselbereichen erweitert: Produktivität, Verfügbarkeit und Integration.

Produktivität:

- Java auf der Datenbank:

Java ist die Sprache des Internets. ASE integriert Java auf der Datenbank und erlaubt dadurch dem Entwickler, Java-Code auf dem Server zu schreiben, zu speichern und auszuführen.

- XML Unterstützung:

Mit Hilfe der XML Unterstützung ist es möglich, Datenbanken zu sichern und wiederherzustellen. Zusätzlich können darüber ständige Sicherungen und Replikationen realisiert werden.

- Web Transaction Management:

ASE sichert die Integrität von verteilten Transaktionen. Der Transaktionsmanager regelt

Zugriffe von verschiedenen Umgebungen mit verschiedenartigen Datenbanksystemen und koordiniert auch Zugriffe über mehrere Sybase Datenbankserver.

Verfügbarkeit:

- fortlaufende Datenbankverfügbarkeit:
Um ungeplante Ausfälle zu vermeiden, unterstützt ASE in einer Zwei-Knoten-Cluster Konfiguration, die Angebote verschiedener Hersteller – wie z.B. HP, SUN, IBM, etc. - von "high-availability" Lösungen. Im Falle eines Ausfalls übernimmt der Stand-by-Server alle Aufgaben und stellt die ausgefallene Datenbank wieder her. Durch den nahtlosen Übergang zwischen den beiden Servern kann eine Verfügbarkeit von fast 100% gewährleistet werden.
- Echtzeit Datenbank-Wartung und –Abstimmung:
Über mitgelieferte Tools können Datenbankadministratoren zur Laufzeit die Datenbank warten und Abstimmungen zur Performanceverbesserung konfigurieren.

Integration:

- Zugang und Analyse gemeinsamer Informationen:
Durch Erstellen einer Proxy-Datenbank und von Tabellendefinitionen können Anwendungen auf Informationen verschiedener Server mit unterschiedlichen Datenbanksystemen zugreifen. Dieses geschieht über einen allgemeinen Datenbankkatalog für Sybase und andere Systeme, der mit dem ASE mitgeliefert wird.
- Plug'N'Play mit standardisierten Internettechnologien:
ASE unterstützt den Zugriff auf standardisierte Internetschnittstellen - wie HTTP und CORBA.

3.2.1.1 Transact SQL

SQL steht für Structured Query Language. SQL wird benutzt, um mit Datenbanken zu kommunizieren. Sie ist die standardisierte Sprache für Datenbank Management Systeme. SQL Aufrufe werden für Aufgaben wie z.B. aktualisieren oder anzeigen von Daten aus einer Datenbank benutzt. Relationale Datenbanken - wie z.B. Oracle, Sybase, Microsoft SQL Server, Access usw. - benutzen alle SQL als Sprache. Jede der aufgeführten Datenbanksysteme hat zusätzlich zu den Standard Befehlen - wie Select, Update, Delete, Insert, Create und Drop - ihre zusätzlichen spezifischen Erweiterungen. Sybase benutzt ebenso wie der SQL Server von Microsoft die Erweiterung Transact SQL. [15]

Transact-SQL enthält die folgenden Erweiterungen:

- Variablen
- IF..ELSE (Bedingte Ausführungen)
- BEGIN..END
- WHILE..
- CURSOR

Variablen

Variablen gelten immer für einen momentanen Prozeduraufruf. Sie existieren nur bis zu dem Zeitpunkt, an dem das nächste "go" oder "send sql" vom Benutzer abgeschickt wird.

Eine Deklaration einer lokalen Variable hat die folgende Syntax:

```
declare {@variable_name} {datatype}
```

Beispiel:

```
declare @intVar int
declare @charVar char(10)
declare @monVar money, @floatVar float
```

Für die Wertzuweisung von lokalen Variablen wird das `select` Statement benutzt:

```
select @intVar=10
oder
select @intVar=count(*) from PI_MATERIAL
```

Lokale Variablen können als Platzhalter in einer **where** Klausel verwendet werden

Beispiel:

```
/* Alle Bestandsschlüssel für einen Sendetag ** ausgeben */
declare @dateSendetag datetime
select @dateSendetag='02/10/2000'
select PW_BESTANDS_SCHLUESSELi from PI_PROGRAMMWOCHE where
PW_SENDETAGd = @dateSendetag
```

Es existieren zwei Wege, um den Inhalt lokaler Variablen anzuzeigen.

1.Weg: Verwendung des `select` Statements:

```
declare @intVar int
select @intVar =2
```

```
select @intVar
```

Durch das `select` Statement wird der Wert von `intVar` als ein Teil des SQL-Datenflusses wiedergegeben. Innerhalb des Client wird das Resultat des `selects` dargestellt wie bei einen row table. Zusätzlich kann das `select` Statement Ausdrücke von der `select list` haben.

2. Weg: Anzeigen einer lokalen Variablen mit dem `print` Statement. Das `print` Statement funktioniert nur mit Variablen des Typs `char`:

```
declare @charMsg char(80)
select @charMsg = "Dies ist ein Test"
print @charMsg
```

Das `print` Statement sendet eine Textmessage zum ErrorMessage-Handler des Clients (obwohl es kein Fehler ist). Der Client behandelt die Ausgabe des `print` Statements auf dem gleichen Weg wie eine Fehlermeldung des Servers (z.B.: in einer PopUp Box).

Das `print` Statement wird normalerweise zum Senden von Warnungen oder Statusmessages zum Endbenutzer genutzt. [3]

IF..ELSE (Bedingte Ausführung)

Die Deklaration einer bedingten Ausführungen hat die folgende Syntax:

```
if {boolean_expression}
    {Statement}
else
    {Statement}
```

Ein Boolean Ausdruck benutzt Standardvergleichsoperatoren, wie z. B.:

`>`, `<`, `=>`, `<=`, oder `!=`. Es können als Vergleichswerte sowohl absolute Werte als auch lokale Variablen verwendet werden.

Beispiel:

```
declare @intAnzahl int
select @intAnzahl =count(*) from PI_SERIE
if @intAnzahl => 20
select "Es sind 20 oder mehr verschiedene Serien in der
Programmwoche eingeplant "
else
```

```
select "Es sind weniger als 20 verschiedene Serien in der
Programmwoche eingeplant"
```

Eine spezielle Version des `if` Statements arbeitet mit dem Befehl `exists`. Er testet, ob eine Reihe von der Anfrage zurückgegeben wird. Es macht keinen Unterschied, welche Spalte im `if exists` Test selektiert ist. [3]

Beispiel:

```
if exists ( select SERIE_KEYi from PI_SERIE where
SERIE_SENDETITELvc is null )
    select "Es gibt noch Serien ohne einen Sendetitel"
```

oder:

```
if not exists (select SERIE_KEYi from PI_SERIE where
SERIE_SENDETITELvc is null )
    select "Alle Serien haben einen Sendetitel"
```

BEGIN..END

Die `begin..end` Struktur erlaubt, eine Reihe von Transact-SQL Statements als "Eins" zu behandeln.

Zum Beispiel: in allen vorangegangenen `if` Beispielen konnte nur ein Transact-SQL Statement ausgeführt werden. Die `if` Konstruktion behandelt alle Statements zwischen `begin` und `end` als wären sie ein einziges Statement:

```
if 1 = 1
    begin
        select ....
        update ....
        print ...
    end /* end of the if */
```

`begin..end` hat keine Bedeutung für die Transaktionsverarbeitung. Es beeinflusst in keinem Fall Übergaben oder Rollbacks. [3]

WHILE..

Die Syntax des `while` Statements ist wie folgt:

```
while {boolean expression}
    Statement
```


Beispiel:

```
declare @intVar int
select @intVar =1
while @intVar > 0
begin
    { misc processing}
    select @intVar= @intVar - 1
end
```

Ein break Statement in einem begin...end beendet die while Schleife.

Ein continue Statement in einem begin...end springt bis zum end und startet der nächste Durchlauf der Schleife. [3]

CURSOR

Mit Hilfe eines Cursors können Abfragen zeilenweise abgearbeitet werden. Es ist eine Schleife auf ein Recordset.

Um einen Cursor zu benutzen, muß dieser zuerst zusammen mit einer Abfrage deklariert werden. [15]

```
declare @intSerieKey int
declare c_GetAlleSerienIDs cursor for select SERIE_KEYi from
PI_SERIE
```

Durch das Öffnen des Cursors wird dieser aktiviert und das Select-Statement ausgeführt.

```
open c_GetAlleSerienIDs
```

Mit fetch springt der Zeiger zur ersten Zeile des Recordsets und der Wert der aktuellen Zeile wird in die lokale Variable gespeichert.

```
fetch c_GetAlleSerienIDs into @intSerieKey
```

```
while 0 = @@sqlstatus
begin
```

```

        select @intSerieKey
        fetch c_GetAlleSerienIDs into @intSerieKey
    end

```

Die Schleife wird durchgeführt bis das Recordset abgearbeitet ist. Mit `fetch` springt der Zeiger immer zur nächsten Zeile und liest den nächsten Wert ein.

Über `close` und `deallocate` wird der Cursor geschlossen und der Speicherplatz wieder freigegeben.

```

    close c_GetAlleSerienIDs
    deallocate cursor c_GetAlleSerienIDs

```

3.2.1.2 Stored Procedure

Es ist möglich, Batches mit SQL Statements zu benutzen und sie in einer Prozedur auf der Datenbank zu lagern. Diese Prozedur wird als *Stored Procedure* bezeichnet. [3]

Das Kommando um eine Stored Procedure zu deklarieren ist:

```

    create proc {procedure name}
    as
        {Statement or block of Statements}

```

Zum Beispiel:

```

    create proc MeinProc
    as
    begin
        select * from PI_PROGRAMMWOCHE
    end

```

Parameter ermöglichen es, Stored Procedures dynamisch zu gestalten. Diese Parameter werden bei der Erzeugung von Stored Procedure definiert und sind dann Teil des Statements.

Die Syntax des `create proc` Kommandos mit Parametern ist:

```

    create proc {proc name}
        ( @{param_name} {param_type},
          @{param_name} {param_type},
          {...}
        )

```

```
as
    {Statement}
```

Zum Beispiel:

```
create proc MeinProc
    ( @dateSendetagVon datetime,
      @dateSendetagBis datetime
    )
as
begin
    select *
    from PI_PROGRAMMWOCHE
    where PW_SENDETAGd between @dateSendetagVon and
        @dateSendetagBis
end
```

Stored Procedures werden wie folgt auf der Datenbank aufgerufen:

Syntax:

```
exec [Datenbank.owner.]{procname} {opt params}
```

Beispiel:

```
exec MeinProc '02/01/2000' , '02/06/2000'
```

Diese Stored Procedure liefert das gleiche Ergebnis wie die Ausführung des folgenden Statements:

```
select *
from PI_PROGRAMMWOCHE
where PW_SENDETAGd between '02/01/2000' and '02/06/2000'
```

Stored Procedures können einen Rückgabewert über das return Statement liefern.

Beispiel:

```
create MeinProc
(
    @dateSendetag datetime
)
as
```

```

begin
select * from PI_PROGRAMMWOCHE
where PW_SENDETAGd = @dateSendetag
  if @@rowcount = 0
      return 1
  else
      return 0
end

```

Der Rückgabewert ist 0 bei erfolgreicher Abarbeitung der Stored Procedure, andernfalls 1.

Beispiel um eine Stored Procedure mit einem Rückgabewert aufzurufen:

```

declare @intStatus int
exec @intStatus = MeinProc '02/01/2000'
if @intStatus != 0
  begin
    {do error processing}
  end
else
  begin
    {do normal processing}
  end
end

```

Die Parameter von Stored Procedures können mit Werten vorbelegt werden:

```

create proc MeinProc
  ( @intVar1 int = 0,
    @intVar2 int = null,
    @charVar3 char(20) = "mydefault"
  )
as ...

```

Beim Aufruf der Procedure müssen nicht mehr alle Übergabeparameter angegeben werden:

```

exec MeinProc 1          /* params 2 & 3 default */

```

Stored Procedures können auch Ausgabeparameter liefern:

```
create proc MeinProc
    ( @dateSendetag    datetime,
      @intAnzahlOut    int output )
as
    select @intAnzahlOut=count(*)
    from PI_PROGRAMMWOCHE
    where PW_SENDETAGd = @dateSendetag
```

Wichtig beim Aufruf einer Procedure mit Ausgabeparameter ist, daß hinter den Variablen, die die Ausgabewerte annehmen sollen, der Aufruf 'out' angegeben wird.

```
declare @Anzahl int
exec MeinProc '02/01/2000', @Anzahl out
select @Anzahl
```

Wenn eine Stored Procedure erstellt wird, speichert der Server deren Text in der Tabelle "syscomments" der aktuellen Datenbank. Zeitgleich generiert und speichert der Server eine normalisierte Form der Procedure, einen sogenannten `query tree`, in die Tabelle `sysprocedures`. Dieser `query tree` wird vom Server benötigt, um einen `query execution plan` zu erzeugen, der beim Ausführen einer Procedure benutzt wird.

Das Erstellen des `query trees` wird auch Resolution genannt und läuft wie folgt ab:

Der Prozeß zerlegt die Procedure in ein effizienteres Format und löst alle beteiligten Objekte in ihre interne Darstellung auf. Dabei werden alle Tabellennamen in ihre Objekt-ID und alle Spaltennamen in ihre Spalten-ID konvertiert.

Das Erstellen des `query plan` wird auch Compilation genannt und läuft wie folgt ab:

Der Server generiert beim ersten Aufruf einer Procedure einen `query plan`. Wenn die Procedure aufgerufen wird, liest der Server den dazugehörigen `query tree` aus der Tabelle `sysprocedures` und lädt ihn in seinen Cache. Der Server legt den optimalen Weg zum Zugriff auf die Daten fest und generiert den `query plan` basierend auf folgenden Informationen:

- das gespeicherte SQL im `query tree`,
- Statistiken für jede Tabelle und Verweise auf Indexe in der Procedure,
- der Wert von jedem übergebenen Parameter der Procedure beim ersten Aufruf.

Dieser Plan wird ebenfalls im Cache des Servers gespeichert. Da er nur dort vorhanden ist, muß er jedesmal neu generiert werden, wenn der Server neu gestartet wird.

Ein `query plan` kann von jedem User benutzt werden, aber nur von einer Person gleichzeitig. Sollten mehrere Benutzer dieselbe Procedure zeitgleich aufrufen, wird für jedes weitere Ausführen ein zusätzlicher `query plan` erzeugt und im Cache gespeichert. Diese Pläne können sich bei einem Aufruf mit anderen Parametern vom ursprünglichen unterscheiden.

Der Datenbankadministrator oder der User hat keine Kontrolle darüber, welcher `execution plan` beim Aufrufen der Procedure benutzt wird. Dadurch können die Ausführungszeiten eines Aufrufs mit gleichen Parametern und dem gleichen Ergebnis unterschiedlich lange dauern. Dieses kann verhindert werden, indem beim Erzeugen oder beim Aufruf einer Procedure das Schlüsselwort `with recompile` angegeben wird. So ist gewährleistet, daß jedesmal ein neuer `query execution plan` generiert und benutzt wird. Diese Methode verringert natürlich die Performance einer Procedure, da bei jedem Aufruf diese neu kompiliert werden muß. [8]

Über die `showplan`-Funktion kann der vom Server beim Ausführen einer Procedure benutzte Plan visualisiert werden. Es wird jeder Schritt angezeigt, der im Plan beschrieben ist, wie z.B. die Joins über Tabellen oder einen Index auslesen.

Diese Funktion muß, bevor die Procedure ausgeführt wird, aktiviert werden. Dieses geschieht über den Aufruf:

```
set showplan on
go
```

Danach wird die Procedure aufgerufen. Als Beispiel wird dafür die eben erstellte Funktion mit folgendem Parameter ausgeführt:

```
declare @Anzahl int
exec MeinProc '12/16/2000', @Anzahl out
select @Anzahl
```

Dieses bewirkt folgende Ausgabe:

```
QUERY PLAN FOR STATEMENT 1 (at line 1).
STEP 1
The type of query is DECLARE.
```

QUERY PLAN FOR STATEMENT 2 (at line 2).

STEP 1

The type of query is EXECUTE.

QUERY PLAN FOR STATEMENT 3 (at line 3).

STEP 1

The type of query is SELECT.

QUERY PLAN FOR STATEMENT 1 (at line 0).

STEP 1

The type of query is DECLARE.

QUERY PLAN FOR STATEMENT 2 (at line 5).

STEP 1

The type of query is SELECT.

Evaluate Ungrouped COUNT AGGREGATE.

FROM TABLE

PI_PROGRAMMWOCHE

Nested iteration.

Index : PW_SENDETAG_Idx

Forward scan.

Positioning by key.

Index contains all needed columns. Base table will not be read.

Keys are:

PW_SENDETAGd ASC

Using I/O Size 2 Kbytes for index leaf pages.

With LRU Buffer Replacement Strategy for index leaf pages.

STEP 2

The type of query is SELECT.

Evaluate Ungrouped ASSIGNMENT OPERATOR.

Erläuterungen zur Ausgabe des showplan:

- `STEP n`
Diese Anweisung liefert der showplan für jede ausgeführte Abfrage. `n` steht für einen Integerwert, beginnend mit 1. Manche Abfragen können vom SQL-Server nicht in einem Schritt bearbeitet werden, deshalb wird diese in mehrere aufgeteilt. In diesem Fall wird auch der Integerwert hoch gezählt.
- `The type of query is SELECT / DECLARE / EXECUTE`
Beschreibt die Aktion, die der SQL-Server gerade durchführt (eine Variable deklarieren, Zeilen aus einer Tabelle selektieren, eine Procedure aufrufen).
- `Nested iteration`
Die `Nested iteration` ist die Standardtechnik für das Joinen von Tabellen und die Zurückgabe selektierter Zeilen einer Tabelle. Es zeigt, daß der SQL-Server einen oder mehrere Schleifen durchläuft, um die Tabelle zu lesen und die Zeilen zu sammeln, die den Suchkriterien der Where-Klausel entsprechen.

Die showplan-Methode ist die einzige Möglichkeit zu prüfen, welchen Weg der `query execution plan` beim Ausführen einer Procedure benutzt. Eine Möglichkeit der graphische Darstellung wird nicht unterstützt. [9]

3.2.1.3 Trigger

Trigger sind einer Stored Procedure ähnlich. Sie werden aufgerufen, wenn Daten in eine Tabelle eingefügt, gelöscht oder auf den neusten Stand gebracht werden sollen.

Ein Trigger kann die versuchte Operation überprüfen und sie rückgängig machen (über `rollbacktransactions`), wenn die Daten referenziell unvereinbar sind oder es sich einfach um falsch "formatierte" Daten handelt. Er ist transparent zum Client der `insert`, `update` oder `delete` Operationen aufruft.

Ein Trigger wird immer vom Server aufgerufen. Einige spezielle Kommandos haben die Eigenschaft, Trigger zu übergehen.

Sie können auch benutzt werden, um duplizierte oder vorberechnete Daten auf dem neusten Stand zu halten.

Für eine ausgesuchte Tabelle kann es separate `insert`, `update` und `delete` Trigger geben. [3]

Die durch einen Trigger geänderten Daten (inserted/updated/deleted) sind aus einer von zwei virtuellen Datenbanktabellen zu lesen:

- inserted
- deleted

für inserts:

Die *inserted virtual table* enthält eine Kopie von Reihen, die eingefügt wurden. Es können verschiedene Reihen in dieser virtuellen Datenbanktabelle sein.

für updates:

Die *deleted virtual table* beinhaltet eine Kopie von den Reihen, welche vom update überschrieben wurden. Die *inserted virtual table* enthält eine Kopie der geänderten Reihen.

für deletes:

Die *deleted virtual table* enthält die Reihen, die gelöscht wurden.

Es kann nur ein Trigger für einen gegebenen Typ (insert/update/delete) auf einer Tabelle geben.

Ein Trigger kann sich nicht selbst wieder durch ein insert/update/delete aufrufen (keine Rekursion).

Beispiel für einen Trigger:

Insert Trigger

```
create trigger ti_PI_SERIE on PI_SERIE
for insert
as
declare @SendetitelNeu varchar(254), @Sendetitel varchar(254),
@Serie_Key int

//Alle gerade eingefügten Serien auslesen
declare c_ti_PI_SERIE cursor for
    select i.SERIE_SENDETITELvc, i.SERIE_KEYi
    from inserted i

open c_ti_PI_SERIE
```

```

fetch c_ti_PI_SERIE into @Sendetitel, @Serie_Key
while 0 = @@sqlstatus
BEGIN

    //PROFIS-Sendetitel wird umkonvertiert für die Ausgabe
    //in der Programmwoche
    exec bereinigeSendetitel @Sendetitel, @SendetitelNeu
output

    //PW-Sendetitel wird in PI_SERIE gespeichert
    update PI_SERIE set SERIE_SENDETITEL_PWvc =
    @SendetitelNeu where SERIE_KEYi = @Serie_Key

    //nächsten Eintrag auslesen
    fetch c_ti_PI_SERIE into @Sendetitel, @Serie_Key
END

close c_ti_PI_SERIE
deallocate cursor c_ti_PI_SERIE

```

3.2.2 ASP

3.2.2.1 Einführung

ASP steht für **Active Server Pages**. Es ist eine Webapplikation mit deren Hilfe interaktive Seiten für das Inter- und Intranet auf einfache Weise erzeugt werden können.

ASP ist eine Alternative zu CGI für Webentwickler, die Perl nicht beherrschen und einen Microsoft-Webserver nutzen. Mit ASP ist es ohne großen Programmieraufwand möglich, interaktive serverseitige Anwendungen zu erzeugen. [14]

Active Server Pages basiert auf Dateien mit der Erweiterung `.ASP`. Eine ASP-Datei ist eine Textdatei, die beliebige Kombinationen der folgenden Elemente enthalten kann:

- Text
- HTML-Kennzeichen
- Skriptbefehle

Ein Skriptbefehl weist den Computer an, eine Aktion auszuführen (z.B. einer Variable einen Wert zuzuweisen).

Erhält der Server eine Seite, die nur HTML-Code enthält, wird diese auch als HTML-Seite an den Browser zurückgegeben und angezeigt.

Die Eigenschaften von ASP zeigen sich erst, wenn die HTML-Dateien um Skripte erweitert werden.

Ein Skript ist eine Folge von Skriptbefehlen. Mit diesen kann man z.B.:

- einer *Variablen* einen Wert zuweisen,
- den Webserver anweisen, ein Element, z.B. den Wert einer Variablen, an einen Browser zu senden,
- Befehle zu *Prozeduren* zusammenfassen.

Durch das Ausführen eines Skripts wird eine Befehlsfolge zu einem Skriptmodul gesendet, welches diese interpretiert und an den Computer überträgt. Skripte werden in Sprachen geschrieben, die bestimmten Regeln unterliegen.

Die Kennzeichen einer Skriptsprache sind:

- sie besitzen im allgemeinen nur ein allgemeinen Datentyp,
- es handelt sich um eine Interpretersprache,
- die Befehle haben eine höhere Komplexität,
- und sie greifen auf Komponenten zurück, die in einer anderen Sprachen entwickelt worden sind. [7]

Gegenüber einer herkömmlichen Programmiersprachen haben Skriptsprachen folgende Vorteil:

- eine einfache Erlernbarkeit,
- Skriptsprachen bieten meist besondere Features an (z.B.: Stringverarbeitung mit regulären Ausdrücken, Socketprogrammierung usw.),
- schnellerer Programmierablauf (keine Kompilierung, Änderungen können sofort getestet werden),
- der Programmtext verringert sich durch die Komplexität der Befehle,
- "ausführbare Datei" deutlich kleiner als entsprechende richtige Programme (Vorteil im Internet),
- die Skripte können während des Programmablaufes erstellt und modifiziert werden ,
- ohne Neukompilierung auf verschiedenen Betriebssystemen einsetzbar (sofern keine systemspezifischen Befehle eingesetzt wurden). [7]

Die Nachteile von Skriptsprachen sind:

- keine Unterstützung von Datenstrukturen,
- und bedingt durch längere Ausführungszeiten nicht für rechenintensive Anwendungen geeignet. [7]

Wenn eine bestimmte Skriptsprache verwendet wird, muss auf dem Server ein Skriptmodul ausgeführt werden, das diese Sprache interpretieren kann. ASP enthält Skriptmodule für die Skriptsprachen VBScript und JScript. Die voreingestellte *primäre Skriptsprache*, d.h. die verwendete Sprache wenn keine andere explizit angegeben wird, ist VBScript.

ASP selbst ist keine Skriptsprache. Es bietet nur eine Umgebung, in der Skripte verarbeitet werden können, die in HTML-Seiten integriert wurden. Um erfolgreich mit ASP arbeiten zu können, muß die zugrundeliegende *Syntax* bzw. die Regeln erlernt werden. [14]

3.2.1.1 Trennzeichen

HTML-Kennzeichen werden vom Text durch *Trennzeichen* abgetrennt. Ein Trennzeichen ist ein Zeichen oder eine Zeichenfolge. Dadurch wird der Anfang oder das Ende einer Einheit markiert. Die HTML-Trennzeichen sind das "Kleiner-als"-Symbol (<) und das "Größer-als"-Symbol (>).

Auch ASP-Skriptbefehle und Ausgabeausdrücke werden vom Text und den HTML-Kennzeichen durch Trennzeichen abgetrennt. ASP verwendet die Trennzeichen <% und %>, mit denen Skriptbefehle umschlossen werden. Der Befehl <% sport = "Klettern" %> weist z.B. den Wert Klettern der Variable sport zu.

ASP umgibt Ausgabeausdrücke mit den Trennzeichen <%= und %>. Der Ausgabeausdruck <%= sport %> sendet z.B. den Wert der Variablen "sport" (in diesem Fall "Klettern") zum Browser. [14]

3.2.1.2 Einzelne Ausdrücke

Mit den ASP-Trennzeichen werden alle Ausdrücke umschlossen, die für die primäre Skriptsprache gültig sind. Die folgende Zeile generiert z.B. einen Text, der die aktuelle Zeit auf dem Server ausgibt:

Diese Seite wurde zum letzten Mal um <%= Now %> aktualisiert.

In diesem Fall gibt der Webserver neben dem Text den Wert der VBScript-Funktion **Now** an den Browser zurück. [11]

3.2.1.3 Anweisungen

Eine *Anweisung* unabhängig ob sie in VBScript oder anderen Skriptsprachen geschrieben wird, ist eine syntaktisch vollständige Einheit, die einen Typ von Aktion, Deklaration oder Definition ausdrückt. Die folgende Bedingungsanweisung **If...Then...Else** ist eine typische VBScript-Anweisung:

```
<%  
If Time >=#12:00:00 AM# And Time < #12:00:00 PM# Then  
    greeting = "Guten Morgen!"  
Else  
    greeting = "Hallo!"  
End If  
>
```

Diese Anweisung speichert entweder den Wert "Guten Morgen!" oder den Wert "Hallo!" in der Variablen `greeting`. Über folgenden Code würde der Clientbrowser den Wert ausgeben:

```
<FONT COLOR="GREEN">  
<%= greeting %>  
</FONT>
```

Daher wird beim Anzeigen dieses Skripts vor 12:00 Uhr (in der Zeitzone des Webserver) folgendes angezeigt:

Guten Morgen!

Um oder nach 12:00 Uhr wird folgendes angezeigt:

Hallo!

Zwischen die Abschnitte einer Anweisung kann ein HTML-Text eingefügt werden. Im folgenden Skript z.B. wird HTML-Text in eine Anweisung vom Typ **If...Then...Else** eingefügt, wodurch dasselbe Ergebnis generiert wird wie mit dem Skript im vorherigen Abschnitt:

```
<FONT COLOR="GREEN">  
<% If Time > = #12:00:00 AM# And Time < #12:00:00 PM# Then  
%>  
Guten Morgen!  
<% Else %>  
Hallo!  
<% End If %>  
</FONT>
```

Wenn die Bedingung erfüllt ist, d.h. wenn der Zeitpunkt um oder nach Mitternacht und gleichzeitig vor 12:00 mittags liegt, sendet der Webserver den HTML-Text im Anschluss an die Bedingung ("Guten Morgen") zum Browser. Andernfalls wird der HTML-Text an den Browser gesendet, der auf **Else** folgt ("Hallo").

Anweisungen, Ausdrücke, Befehle und Prozeduren, die mit Skripttrennzeichen umschlossen sind, müssen in der voreingestellten primären Skriptsprache zulässig sein. Die voreingestellte Skriptsprache von ASP ist VBScript. In ASP können jedoch auch andere Skriptsprachen verwendet werden. Um vollständige Prozeduren einzuschließen, die in einer Sprache geschrieben wurden, für die ein Skriptmodul vorhanden ist, müssen die HTML-Skriptkennzeichen <SCRIPT> und </SCRIPT> zusammen mit den Attributen LANGUAGE und RUNAT eingesetzt werden. [11]

Die folgende ASP-Datei verarbeitet z.B. die JScript-Funktion **Funktion1**.

```
<HTML>
<BODY>
<% Call Funktion1 %>
</BODY>
</HTML>

<SCRIPT RUNAT=SERVER LANGUAGE=JSCRIPT>
    function Funktion1 ()
    {
        Response.Write("Funktion1 aufgerufen")
    }
</SCRIPT>
```

3.2.1.4 Datenbankzugriff

Für Datenbankzugriffe werden Active Data Objects (ADO) für den unkomplizierten Zugriff auf die Daten verwendet.

Es kann sich dabei um Server-Datenbanken von einem beliebigen Hersteller handeln, solange ein ODBC - oder OLE-DB-Treiber zur Verfügung steht, auf den das Objekt zugreifen kann. [4]

ODBC (Open Database Connectivity) ist eine von Microsoft entwickelte standardisierte Anwendungs-Programmierungs-Schnittstelle (Application Programming Interface, API). Anwendungen können über diese Schnittstelle auf verschiedene Datenquellen zugreifen, für

die es ODBC-kompatible Treiber gibt. Als Standardsprache zum Datenzugriff wird dabei SQL verwendet.

Mit Hilfe von ODBC können Applikationen auf vorhandene Datenelemente zugreifen, welche innerhalb von DBMS-Systemen auf PC's, Workstations und Mainframes abgelegt sind. Sie sind außerdem völlig unabhängig von den unterschiedlichen Datenformaten der bestehenden Datenbanksysteme (z.B. Sybase, Oracle), sowie ihrer zugehörigen Programmierschnittstellen.

[5]

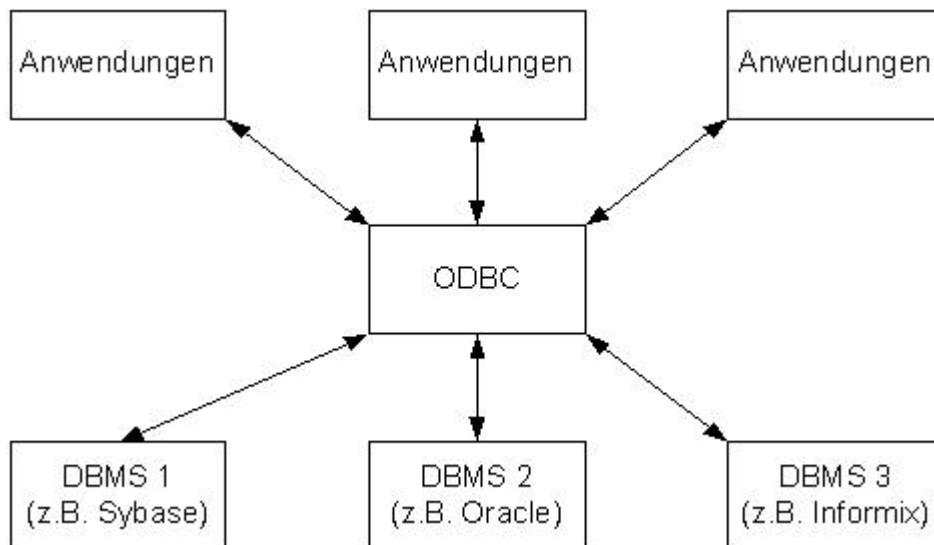


Abbildung 1 DB-Aufrufe mit ODBC

In einer ODBC-API wird folgendes definiert:

- eine Bibliothek mit ODBC-Funktionsaufrufen, mit denen eine Verbindung zur Datenquelle hergestellt, SQL-Anweisungen ausgeführt und Ergebnisse abgerufen werden;
- eine auf der X/Open und SQL Access Group (SAG) CAE-Spezifikation (1992) aufbauende SQL-Syntax;
- eine standardisierte Darstellungsweise von Datentypen;
- eine standardisierte Gruppe von Fehlercodes.

Die ODBC-Architektur besteht im Wesentlichen aus vier Komponenten:

- Treiber

Der Treiber verarbeitet die Funktionsaufrufe. Er übermittelt die SQL-Anfragen an die jeweilige Datenquellen und liefert die Ergebnisse an die Anwendungen zurück.

- Treiber-Manager

Der Treiber-Manager lädt die angeforderten Treiber, die von der Applikation benötigt werden.

- Applikationsebene

Die Applikationsebene leitet die Verarbeitung und die Aufrufe der Funktion. Sie überträgt SQL-Statements an die Treiber und empfängt die Ergebnisse.

- Datenquelle

Die Datenquellen beinhalten die Daten. Die daraus resultierenden Datenbankzugriffe können über die unterschiedlichsten Netzwerkverbindungen und von verschiedensten Plattformen erfolgen.

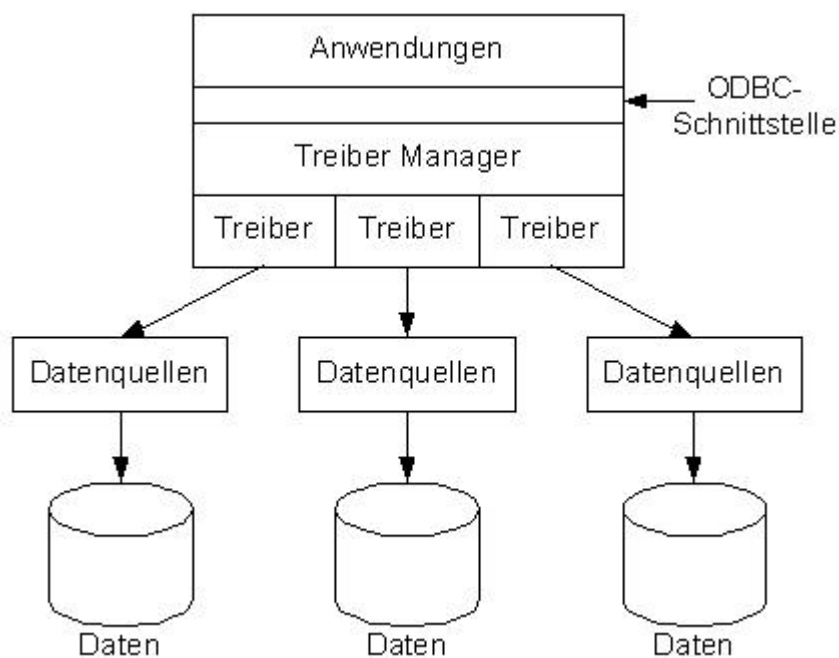


Abbildung 2 Die ODBC-Architektur

OLE-DB ist eine auf dem Component Objekt Model-Standard (COM) basierender Nachfolger für Datenbankzugriffe. Es wurde ebenfalls von Microsoft entwickelt und ist der Nachfolger von ODBC, wobei eine starke Integration geschaffen wurde, welche die Verwendung bisheriger Treiber weiterhin erlaubt. Das besondere an OLE-DB ist, daß es für beliebige Daten (RDBMS, E-Mail, etc.) konzipiert wurde. [6]

Die Merkmale von OLE-DB sind:

- die Definition der Standardschnittstellen,

- die drei Arten von Komponententypen:
 - Data Provider: Darstellung von Daten als virtuelle Tabelle
 - Data Consumer: nutzt Dienste eines Data Providers
 - Service Component: Kapselung einer DBMS-Funktion (Abfrage-Prozessoren, Cursor-Modelle, Transaktions-Manager).

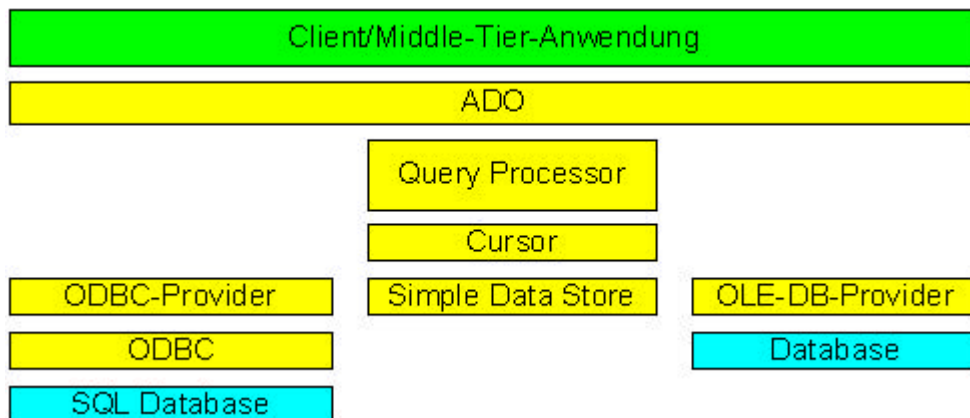


Abbildung 3 Aufbau von OLE-DB

Damit ein Active Data-Objekt verwendet werden kann, wird zuerst - wie gewohnt - eine Instanz erstellt. [4]

Dafür gibt es den folgenden Script-Befehl:

```
<%
```

```
Set DatabaseConnection =
Server.CreateObject("ADODB.Recordset")
```

Danach wird dem Objekt der Connection-String übergeben, mit dem es sich an der Datenbank anmeldet. In diesen String sind enthalten: der verwendete Treiber, OLE-DB-Datenquelle, Datenbankname, Benutzername und Paßwort.

```
DatabaseConnection.ActiveConnection =
"Provider=Sybase.ASEOLEDBProvider.2;Persist Security
Info=False;Data Source=ProgrammwocheOle;Initial
Catalog=Programmwoche;pwd=Urban;User ID=Egal;"
```

Mit der Methode **Source** wird die **SELECT**-Anweisung an die Datenbank deklariert, die später abgeschickt wird.

```
SQLQuery = "SELECT PW_NUMMERi, PW_STARTZEITd FROM
PI_PROGRAMMWOCHE"
DatabaseConnection.Source= SQLQuery
```

Über die Methode **Open** wird die Abfrage auf die Datenbank gestartet und das Recordset zum späteren Auswerten gefüllt.

```
DatabaseConnection.Open
```

```
%>
```

Die Ergebnismenge ist eine Tabelle mit Werten, deren Struktur von den Feldern bestimmt wird, die in der SQL **SELECT**-Anweisung spezifiziert wurden. Das Anzeigen der von der Abfrage zurückgegebenen Zeilen entspricht daher dem Durchführen einer Schleife durch die Zeilen der Ergebnismenge. In diesem Beispiel werden die zurückgegebenen Daten in den Zeilen einer HTML-Tabelle angezeigt.

```
<% Do While Not DatabaseConnection.EOF %>
```

```
<TR>
```

```
<TD BGCOLOR="f7efde" ALIGN=CENTER>
```

```
<FONT STYLE="ARIAL NARROW" SIZE=1>
```

```
<%= DatabaseConnection("PW_NUMMERi") %>
```

```
</FONT></TD>
```

```
<TD BGCOLOR="f7efde" ALIGN=CENTER>
```

```
<FONT STYLE="ARIAL NARROW" SIZE=1>
```

```
<%= DatabaseConnection("PW_STARTZEITd") %>
```

```
</FONT></TD>
```

```
</TR>
```

Wenn eine Bedingung wahr ist, wiederholt die **Do...Loop**-Anweisung einen Block von Befehlen. Bei diesen kann es sich um Skriptbefehle bzw. HTML-Text und -Kennzeichen handeln. Im Beispiel wird innerhalb der Schleife über HTML-Kennzeichen eine Tabellenzeile mit zwei Spalten generiert. Über Skriptbefehle werden die Werte auf der aktuellen Position des Recordsets ausgelesen und in die Felder der Tabelle geschrieben.

Zum Beenden der Schleife wird die **MoveNext**-Methode verwendet, um den Zeilenzeiger für die Ergebnismenge eine Zeile nach unten zu verschieben. Da diese Anweisung noch zur **Do...Loop**-Anweisung gehört, wird sie wiederholt, bis das Ende des Recordsets erreicht ist.

```
<%
```

```
DatabaseConnection.MoveNext
```

```
Loop
```

```
%>
```

4 Ist-Analyse

4.1 Programmwoche

Die Programmwoche wird als Print- und Onlineversion wöchentlich zur internen und externen Verwendung erstellt.

Der Prozeß der Erstellung beginnt mit der Teilnahme eines Mitarbeiters der Abteilung Presse/Programminformation an der Programmplanungssitzung. Diese findet jeden Dienstag statt. Dort wird der Ablaufplan besprochen, der in 6 Wochen ausgestrahlt werden soll. Als Ergebnis dieser Besprechung erhält die Abteilung eine Word-Datei von der Programmplanung, die eine Rohfassung des Programmablaufs enthält. Diese Datei wird jetzt in ein bestimmtes Format gebracht, damit später daraus die HTML-Seiten generiert werden können. Dafür werden nicht benötigte Informationen aus dem Word-Dokument gelöscht. Danach wird das vorhandene Informationsmaterial zu den einzelnen Sendungen gesichtet. Bei Wiederholungen sind in der Regel schon Texte im Programmwochenarchiv vorhanden und können mit nur kleinen Änderungen übernommen werden. Wenn noch kein Material vorhanden ist, wird über die Sendung recherchiert und ein komplett neuer Text erstellt. Die Texte werden nun in dem Dokument zusammengestellt und dienen als Vorlage für den Langablauf. Aus dieser Programmwoche wird nun das Highlight der Woche ausgewählt. Dafür werden weitere Hintergrundinformationen gesammelt, Bilder bereitgestellt und in einem gesonderten Word-Dokument zusammengefaßt. Danach wird der Kurzablaufplan erstellt, der nur die Startzeiten und die Namen der Sendungen enthält. Diese Informationen werden aus dem zuvor erstellten Langablaufplan entnommen und ebenfalls als eigenständiges Dokument gespeichert. Jetzt werden noch die aktuellen und noch nicht publizierten Änderungen der 5 vorangegangenen Wochen zusammengefaßt. Nun beginnt die Recherche für die Spielfilmvorschau der kommenden Woche (7. Woche). Die Informationen über die Filme, wie Sendedatum, Startzeit usw., kommen dabei aus der Filmredaktion. Diese Vorschau wird ebenfalls in einem Dokument gespeichert. Danach erfolgt noch einmal ein Abgleich mit der Programmplanung um eventuelle Änderungen einbringen zu können.

Die erstellten Dokumente werden nun Online, per Fax, gedruckt oder per E-Mail an die externen Instanzen wie Programmpresse, Tageszeitungen, Text-/ Bildagenturen, Radio, Produktionsfirmen und IPA und internen Abteilungen wie Bild-, Film-, Onlineredaktion, Programmbereitstellung und Presseredaktion versendet. Über ein Wandlungsprogramm werden die Word-Dateien in HTML konvertiert und ins Intranet gespielt. Die jeweils veraltete

Programmwoche wird archiviert. Einmal veröffentlichte Programmwochen werden nicht mehr geändert. Änderungen werden immer separat in digitaler und in Papierform abgelegt.

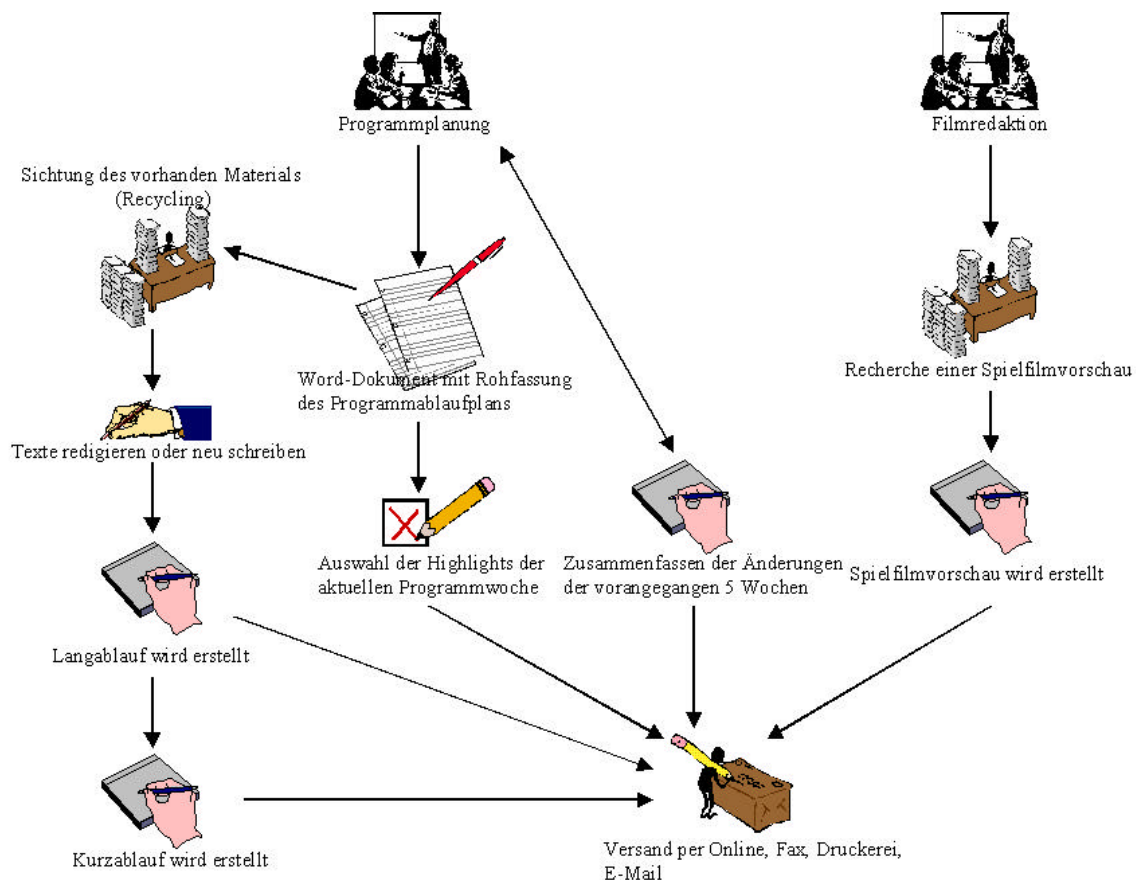


Abbildung 4 Erstellung der Programmwoche

4.2 Bildwoche

Die Bildwoche benutzt die Anwendung Picturedesk, um das Bildmaterial zur Verfügung stellen zu können. Darüber werden die ausgewählten Bilder in der Galerie der Bildwoche im Verzeichnis der aktuellen Bildwoche auf einen Fileserver abgelegt. Die Picturedesk Serversoftware generiert daraus für jedes Bild drei verschiedene Formate: ein Reproformat in Druckqualität, ein Layout- und Thumbnail-Format für die Vorschau auf der HTML-Seite. Zusätzlich wird ein File erstellt, das sämtliche Textinformationen zu den Bildern enthält. Um die Bilder für Kunden zugänglich zu machen, werden diese über das Internet zur Verfügung gestellt. Dieses geschieht über ein extra VB-Client. Dieses Programm greift auf das Filesystem der Bildwoche zu und überprüft, ob alle drei Formate der Bilder, die im Textfile beschrieben werden, vorhanden sind. Danach schreibt es für jedes einzelne Bild ein eigenes Infofile. Über Templates werden die HTML-Seiten generiert und in ein entsprechendes Verzeichnis des Webservers kopiert.

Über einen Web-Browser kann der Kunde sich die Bilder in Thumbnail- oder Layout-Qualität ansehen. Es können Einzelbilder, eine Bildauswahl oder das Bildpaket der kompletten Woche

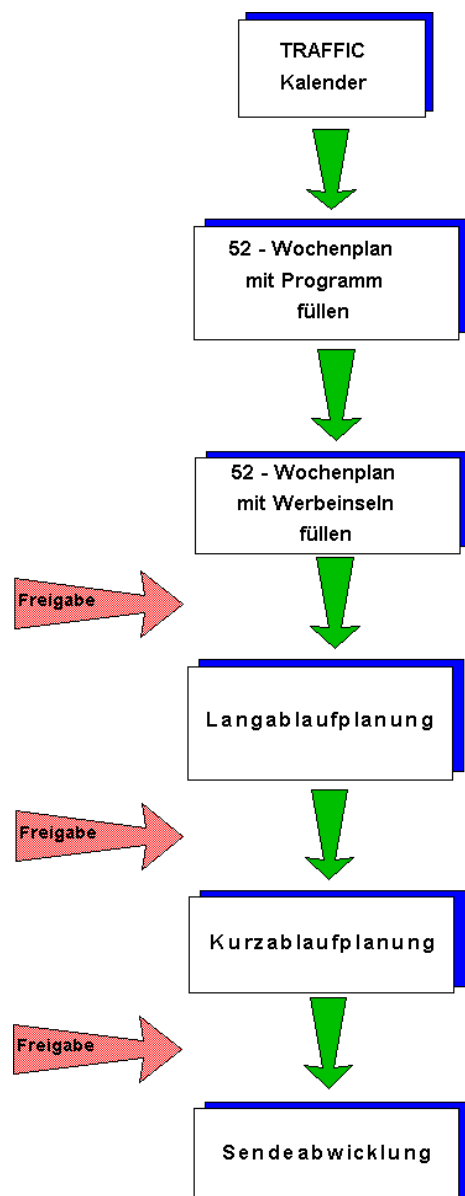
im Reformat heruntergeladen werden. Die Auswahl wird zusammen mit den entsprechenden Infofiles, die die Bildinformationen enthalten, in einem Downloadverzeichnis abgelegt und kann per FTP vom Kunden heruntergeladen werden.

4.3 Schnittstellenbeschreibung zu PROFIS und TRAFFIC

4.3.1 Planung des Sendeablaufs

Zunächst wird ein Kalender in TRAFFIC angelegt. Dabei ist zu beachten, daß es neben den normalen Sendetagen auch Feier- und Brückentage oder Sonderereignisse gibt, die ein eigenes Programmschema haben. In dieser Phase wird ein grober Programmplan mit Platzhaltern für Filme, Serien und Sonderabläufe angelegt. Dieses geschieht schon über 1 Jahr vor der Ausstrahlung eines Sendetages.

Danach beginnt die Planung des 52-Wochenplanes, in der ein Grundgerüst die Programmbestandteile eingefügt wird: die zu laufende Serie wird bestimmt, bei Filmen wird unterschieden, ob eine Kauf- oder Eigenproduktion ausgestrahlt werden soll. Parallel dazu wird das Werbeinselschema hinzugefügt. Damit ist die grobe Planung abgeschlossen. Circa 8 Wochen vor Ausstrahlung beginnt die Langablaufplanung. In der bisherigen Planung wurden Platzhalter für Filme, Serien und Sonderabläufe eingesetzt, die den Sendeplatz reservieren und grob den Programmablauf darstellen. In dieser Phase werden den Sendeplätzen Episoden und Filme aus dem PROFIS-Bestand zugewiesen. Ist dieses abgeschlossen, wird der Sendetag in den Status Kurzablaufplanung gesetzt. Diesen Status behält der Sendetag bis kurz vor der Ausstrahlung (ca. 12 Stunden davor). In dieser Phase wird der Sendetag so geplant, daß es keine Lücken oder Überlappungen zwischen den einzelnen Sendungen mehr vorhanden sind.



Sobald diese Planung abgeschlossen ist, wird der Status auf "Sendefertig" gesetzt. Am Tag der Ausstrahlung übernimmt die Sendeautomation diese Informationen und spielt das geplante Programm ab.

4.3.2 Vorhandene Daten für die Programmwoche

Um eine Programmwoche auf Datenbankbasis zu generieren, ist es wichtig zu wissen, welche Informationen in den Basissystemen vorhanden sind und übernommen werden können.

Die folgenden Abbildungen zeigen Ausschnitten aus dem Programmwochenheft. Dargestellt sind ein Film, eine Episode und ein Sonderablauf mit allen Informationen, die für die jeweilige Sendungsart benötigt werden. Alle Attribute müssen sich auf einer Programmwochendatenbank auch wiederfinden.

02.00 **3** Unter Verdacht **4**

2 (Kriminalfilm, Großbritannien 1991) 95 Minut **5 6 7 8**
 Originaltitel: Under Suspici **9**
 RTL-Erstsendung: 26.05 **13**

Das mondäne Seebad Brighton in den 50er-Jahren: Privatdetektiv und Ex-Polizist Tony Aaron hat sich mit seiner Frau darauf spezialisiert, in Scheidungsfällen "behilflich" zu sein.. **14**

Weihnachten '59 in Brighton. Tony Aaron verdient sich sein Geld als Privatdetektiv auf nicht ganz saubere Weise, nachdem er wegen einer Affäre mit Folgen mehrerhaft aus dem Polizeidienst entlassen wurde.. **15**

"Spannender, raffiniert konstruierter Kriminalfilm, der Elemente der 'Schwarzen Serie' Hollywoods souverän in die englische Gesellschaft der fünfziger Jahre transportiert und seine düstere Grundstimmung bis zum Schluss steigert" (Zoom) **16**

17 Liam Neeson... Tony Aaron **18**
 Laura San Giacomo... Angeline
 (bek. u.a. aus dem Kinohit "Sex, Lügen und Videos" von Steven Sonderbergh und den Filmen "Pretty Woman" und "Quigley, der Australier" mit Tom Selleck)
 Kenneth Cranham... Frank
 (bek. u.a. aus "Prosperos Bücher", Peter Greenaways filmischer Adaption von Shakespeares **19** "Der Sturm")
 Maggie O'Neill... Hazel Aaron
 Alphonsia Emmanuel... Selina
 (bek. u.a. aus der Miniserie "Denver - Die Entscheidung")
 Alan Talbot... Powers
 Malcolm Storry... Waterston
 Martin Grace... Colin
 Kevin Moore... Barrister
 Richard Graham... Denny
 Stephen Moore... Roscoe
 Alan Stocks... Paul
 u.a.; Buch und Regie: Simon Moore **20 21**

Abbildung 5 Auszug aus dem Programmwochenheft für einen Film

Willkommen zu Hause ①

12.15 ② ③ Boston College ④
 (Sitcom, USA 1997) ⑤ ⑥ ⑦

⑨ Originaltitel: Boston Common
 "Joy liebt Boyd" (Mercury Retrograde) ⑩ ⑪

Joy verwechselt aus Versehen Jacks Namen mit Boyds und sieht diesen Versprecher als Zeichen, sich über ihre Beziehung Gedanken zu machen... ⑭

Während einer Unterhaltung mit Jack verwechselt Joy seinen Namen und spricht Jack mit Boyd an. Der gerät völlig außer Fassung, und Joy sieht es als Zeichen, endlich herauszufinden, wen sie nun wirklich liebt. Währenddessen verabredet Boyd sich allerdings mit Lisa zu einem Blind Date, denn Joy hat er schon lange aufgegeben. Als Joy Boyd nun ihre Entscheidung mitteilt, mit ihm eine Beziehung anfangen zu wollen, ist er so verwirrt, dass er beginnt an seiner großen Liebe zu zweifeln. Und obwohl es sein sehnlichster Wunsch ist mit Joy zusammen zu sein, rät er Joy, nochmal über ihre Entscheidung nachzudenken, und trifft sich mit Lisa, seinem Blind Date. Doch bevor Joy sich ernsthaft Gedanken über Boyds Worte machen kann, macht Jack ihr einen Heiratsantrag, und sie willigt ein. ⑮

⑰ Anthony Clark Boyd Pritchett ⑱
 Hedy Burrell Wyleen Pritchett
 Traylor Howard Joy Byrnes
 Steve Paymer Leonard Prince
 Tasha Smith Tasha King
 Vincent Ventresca Jack Reed
 Jennifer Leigh Edwards Lisa
 Regie: Rod Daniel; Buch: Kari Lizer ⑳ ㉑

Abbildung 6 Auszug aus dem Programmwochenheft für eine Serie

18.00 ② **Guten Abend RTL** ③
 Das Ländermagazin ④
 ②① Moderation: Laura Lange

Tagesaktuelle Infos zur Sendung auf RTL-Text S
 650 ①

oder **Regionalprogramme**

Abbildung 7 Auszug aus dem Programmwochenheft für einen Sonderablauf

Die folgende Übersicht stellt die Attribute der Programmwoche den vorhandenen Informationen aus TRAFFIC und PROFIS gegenüber. Die Nummerierung bezieht sich dabei auf die zuvor gezeigten Darstellungen. Neben der Bezeichnung und einer Beschreibung der einzelnen Attribute, die für die Programmwoche notwendig sind, wird beschrieben, wo diese Informationen in den Basissystemen zu finden sind.

Nr.	Bezeichnung	Beschreibung	Datenbank
1	Label	Hinweis auf eine Reihe in Programm, Videotexttafel oder Telefonhotline	Videotext- und Hotlinenummerinformationen sind nicht auf der Datenbank vorhanden.
2	Startzeit	Beginn der Sendung	DB: TRAFFIC TABELLE: Planungsobjekt Spalte: SollStartZeit

3	Sendetitel	Titel des Films, der Serie oder des Sonderablaufs	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Sendetitel
4	Tonformat	Ausstrahlung der Sendung in Mono, Stereo oder Dolby Surround	DB: TRAFFIC TABELLE: Event Spalte: GPI
5	Genre	Kategorie, in der die Sendung einzuordnen ist.	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Genre
6	Produktionsland	Herstellungsland der Sendung	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Produktions_Land
7	Produktionsjahr	Produktionszeitraum der Sendung	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Produktions_Jahr
8	Minutenzahl	Länge der Sendung in Minuten	DB: TRAFFIC TABELLE: Event Spalte: Laenge
9	Originaltitel	Titel der Sendung im Produktionsland	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Originaltitel
10	Episodentitel Deutsch	Titel der Folge einer Serie in deutsch	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Sendetitel
11	Episodentitel Original	Titel der Folge in ihrem Produktionsland	DB: PROFIS TABELLE: Bestandsobjekt Spalte: Originaltitel
12	RTL_Altersempfehlung	Altersempfehlung für Sendungen im Kinderprogramm	Nicht vorhanden
13	Erstausstrahlung	Datum der ersten Ausstrahlung bei RTL	DB: PROFIS TABELLE: sb_ausgestrahlter_Sendebeitrag Spalte: SAS_AUSSTRAHL_DATUM
14	Kurzinhalt	Kurze Zusammenfassung über den Inhalt einer Episode oder eines Films	Nicht vorhanden
15	Langinhalt	Ausführlichere Inhaltsangabe des Films oder der Episode	Nicht vorhanden
16	Zusatztext	Anekdoten über Produktionsnotizen, Besonderheiten während der Dreharbeiten, Zwischenfälle etc.	Nicht vorhanden
17	Schauspieler	Namen der Schauspieler	Werden zwar innerhalb von PROFIS verwaltet, sind aber in dieser Form für eine weitere Verarbeitung nicht nutzbar

18	Rolle	Namen der dargestellten Personen in der Sendung	Werden zwar innerhalb von PROFIS verwaltet, sind aber in dieser Form für eine weitere Verarbeitung nicht nutzbar
19	Kurzvorstellung	Zusammenfassung über die bisherigen Filme des Schauspielers oder Anekdoten aus seinen Leben	Nicht vorhanden
20	Funktion	Funktion innerhalb der Sendung	Werden zwar innerhalb von PROFIS verwaltet, sind aber in dieser Form für eine weitere Verarbeitung nicht nutzbar
21	Name	Name der Person	Werden zwar innerhalb von PROFIS verwaltet, sind aber in dieser Form für eine weitere Verarbeitung nicht nutzbar

4.3.3 Datenbankbeschreibung

Um aus den Basissystemen das Grundgerüst der Programmwoche zu generieren, werden nicht die kompletten Datenbanken von TRAFFIC und PROFIS benötigt. Die folgende Darstellung zeigt, welche Tabellen für die Erstellung einer Programmwochendatenbank benötigt werden und in welcher Beziehung sie miteinander stehen.

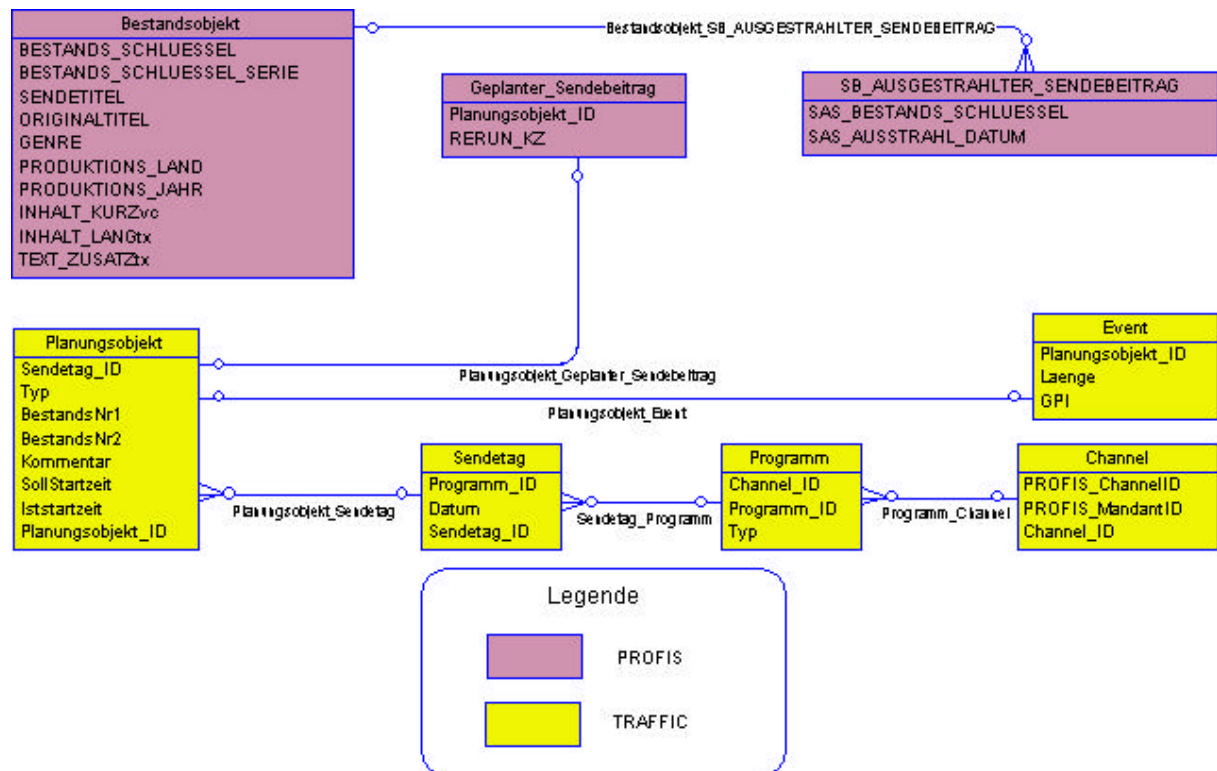


Abbildung 8 Datenbankschnittstelle PROFIS/TRAFFIC

Im folgenden werden die einzelnen Tabellen mit ihrer Herkunft und ihren Feldern beschrieben.

Um den richtigen Programmplan für einen Sender aus TRAFFIC zu bekommen, muß für die Sender- und Mandanten-ID die passende Channel-ID für das TRAFFIC-System besorgt werden. Dies geschieht über die Tabelle *Channel*.

Channel	
Felder	Bemerkung
PROFIS_ChannelID	Referenz zu angegebenem Sender
PROFIS_MandantID	Referenz zu angegebenem Mandanten
Channel_ID	Referenz zu Tabelle Programm

In der Programmplanung werden Änderungen nie direkt in den Programmplan eingegeben. Zuerst werden diese in sogenannten Spielprogrammen eingeplant, um die Auswirkungen auf des restliche Programm zu prüfen. Danach erst werden die Änderungen ins Normalprogramm übernommen. Um den Programmplan zu erhalten, der später ausgestrahlt werden soll, wird die Programm-ID des Normalprogramms und des entsprechenden Channels benötigt. Diese Information ist in der TRAFFIC-Tabelle *Programm* zu finden. Das Normalprogramm hat als Typ immer den Wert 1.

Programm	
Felder	Bemerkung
Channel_ID	Referenz zu Tabelle Channel
Programm_ID	Referenz zu Tabelle Sendetag
Typ	1 für Normal-Programm

Mit Hilfe der Programm-ID und dem Sendetag kann in der Tabelle *Sendetag* auf der TRAFFIC-Datenbank die Sendetag-ID gefunden werden.

Sendetag	
Felder	Bemerkung
Programm_ID	Referenz zu Tabelle Programm
Datum	Sendetag
Sendetag_ID	Referenz zu Tabelle Planungsobjekt

Über diese Sendetag-ID kann der Programmablaufplan eines ganzen Tages gefunden werden. Dies geschieht über die TRAFFIC-Tabelle *Planungsobjekt*. In dieser Tabelle sind alle Sendungen gespeichert, die an einem Tag ausgestrahlt werden sollen, inklusive aller Werbungen und Trailer. Deshalb muß bei einer Abfrage für einen Sendetag über das Feld 'Typ' der Suchbereich auf Filme, Episoden und Sonderabläufe eingegrenzt werden. Eine

weitere Ausnahme stellt das Notprogramm da, welches zusätzlich eingeplant wird, falls z.B. Live-Sendungen frühzeitig enden sollten. Solche Sendungen werden in der Programmwoche nicht ausgegeben. Sie sind daran zu erkennen, daß die Iststartzeit hinter der Startzeit des Sendungstyps 'Sendetage' liegen. Über die beiden Bestandsnummern wird ein Verweis auf die Bestandsobjekte in PROFIS gegeben. Dabei enthalten die Felder folgende Einträge für den jeweiligen Sendungstyp:

- Film: Bestandsnr1 enthält den Schlüssel für ein Bestandsobjekt
- Sonderablauf: Bestandsnr2 enthält den Schlüssel für ein Bestandsobjekt
- Serie: Bestandsnr1: Serienschlüssel, Bestandsnr2: Episodenschlüssel

Diese Tabelle enthält mit der Sollstartzeit und dem Kommentar die ersten Informationen für die Programmwoche.

Planungsobjekt	
Felder	Beschreibung
Sendetag_ID	Referenz zu Tabelle Sendetag
Typ	Aus Wertebereich (Film, Episode, Sonderablauf oder Sendetage)
Bestandsnr1	Referenz zu Tabelle Bestandsobjekt aus PROFIS
Bestandsnr2	Referenz zu Tabelle Bestandsobjekt aus PROFIS
Kommentar	Für Sonderabläufe
SollStartzeit	Startzeit des eingeplanten Objektes
Iststartzeit	Vergleich, ob die Iststartzeit eines Objektes vor dem des Sendetages ist (Ausschluß des Notprogramm)
Planungsobjekt_ID	Referenz zu Tabelle geplanter_Sendebeitrag aus PROFIS und Event aus TRAFFIC

Mit Hilfe der Planungsobjekt-ID aus der Planungsobjekt-Tabelle kann in der TRAFFIC-Tabelle *Event* die Länge und die Vertonung einer Sendung ausgelesen werden. Dies sind wiederum Informationen, die in der Programmwoche benötigt werden.

Event	
Felder	Bemerkung
Planungsobjekt_ID	Referenz zu Tabelle Planungsobjekt
Laenge	Länge einer Sendung
GPI	Vertonung einer Sendung (Mono, Stereo, ...)

Über die Bestandsschlüssel aus den Planungsobjekt können die entsprechenden Objekte aus der PROFIS-Tabelle *Bestandsobjekt* gefunden werden. Diese Tabelle enthält viele Informationen, die zur Erstellung einer Programmwoche benötigt werden. Für jede Sendung werden hier der Sendetitel, Originaltitel, Genre, Produktionsland und Produktionsjahr gespeichert. Ebenfalls ist hier für Episoden ein Verweis auf die Serie abgelegt, der auch in dieser Tabelle zu finden ist.

Bestandsobjekt	
Felder	Bemerkung
BESTANDS_SCHLUESSEL	Referenz zu Tabelle Planungsobjekt aus TRAFFIC Eindeutige Kennung für einen Film, Serie oder Sonderablauf
BESTANDS_SCHLUESSEL_SERIE	Referenz zu Tabelle Bestandsobjekt für eine Serie (nur notwendig, wenn es sich um eine Episode handelt)
SENDETITEL	PROFIS-Sendetitel. Unterscheidet sich zum Sendetitel in der Programmwoche, weil ein Artikel im Titel nachgestellt wird („Wache, Die“)
ORIGINALTITEL	PROFIS-Originaltitel. Gleiche Formatierung wie beim Sendetitel
GENRE	String, der alle Genres einer Sendung beinhaltet.
PRODUKTIONS_LAND	String, der alle Produktionsländer einer Sendung beinhaltet.
PRODUKTIONS_JAHR	String mit von/bis-Angabe

In der PROFIS-Tabelle *Geplanter_Sendebeitrag* kann über die Planungsobjekt-ID das Rerun-Kennzeichen zu einer Sendung ausgelesen werden. "Reruns" sind Wiederholungen eines Filmes oder einer Episode innerhalb von 24 Stunden. Dieses Kennzeichen wird nicht direkt in der Programmwoche ausgegeben, wird aber später in der Programmwochendatenbank benötigt.

Geplanter_Sendebeitrag	
Felder	Bemerkung
Planungsobjekt_ID	Referenz zu Tabelle Planungsobjekt aus TRAFFIC
RERUN_KZ	Kennzeichen für ReRun

Für jede Sendung muß, falls vorhanden, das Erstausstrahlungsdatum mit ausgegeben werden. Diese Information steht in der PROFIS-Tabelle *SB_AUSGESTRAHLTER_SENDEBEITRAG*.

Jeder Ausstrahlungstermin einer Sendung wird hier gespeichert. Um das Erstausstrahlungsdatum zu erhalten, muß das niedrigste Datum für eine Sendung über den Bestandsschlüssel abgefragt werden.

SB_AUSGESTRAHLTER_SENDEBEITRAG	
Felder	Bemerkung
SAS_BESTANDS_SCHLUESSEL	Referenz zu Tabelle Bestandsobjekt
SAS_AUSSTRAHL_DATUM	Datum der Erstausstrahlung einer Sendung

5 Soll-Analyse

5.1 Anforderungen der neuen Programmwoche

5.1.1 Programmwoche

Die neue Programmwoche hat als Vorbild den Langablauf der gedruckten Version. Alle Informationen, die in diesen Plan vorhanden sind, sollen auch auf der Datenbank wiederzufinden sein.

Die Programmwochendatenbank soll die Programmablaufpläne von bis zu 8 Wochen enthalten können. Davon sind 6-7 Wochen von außen über das Internet frei zugänglich. Eine weitere Woche wird im voraus bearbeitet, um sie zu einem späteren Zeitpunkt veröffentlichen zu können.

Die Grundinformationen stellen dabei aus den Basissystemen TRAFFIC und PROFIS zur Verfügung.

Die Informationen einzelner Sendungen sollen über einen Client bearbeitet werden können und in die Programmwochendatenbank zurückgespeichert werden. Die redaktionellen Daten aus Kurzinhalt, Langinhalt und Zusatztext sollen für Filme und Episoden an die PROFIS-Datenbank weitergegeben werden und somit wiederverwendbar sein. Dafür muß die Bestandsobjekttabelle erweitert werden.

Die Programmwoche ist eine eigenständige Datenbank, die nicht von den Systemen TRAFFIC oder PROFIS abhängig ist. Sollten die Basissysteme ausfallen, muß es immer noch möglich sein, eine Programmwoche zu erstellen und zu veröffentlichen.

Die Programmwochendatenbank soll die Programmablaufpläne verschiedener Sender verwalten können. Dafür muß die Datenbank zwischen verschiedenen Mandanten unterscheiden können.

Damit die Verlage mit den Daten arbeiten können, wird es die Möglichkeit geben, die aktuellen Programmpläne als XML- oder Text-Datei herunterzuladen. Mit diesem File können die Verlage wie bisher ihre Programmzeitschriften erstellen.

Eine weitere Anforderung ist die Möglichkeit, sich eine Historie einer bereits veröffentlichten Programmwoche anzusehen. Dies soll über eine Versionierung geschehen. Der Kunde kann, anhand der verschiedenen Versionen der Programmablaufpläne, Änderungen nachvollziehen.

5.1.2 Änderungen

Änderungen sollen, soweit es geht, automatisch erkannt werden. Dafür wird die letzte Version der Programmwochendatenbank mit dem aktuell eingeplanten Stand aus TRAFFIC und PROFIS verglichen. Alle Änderungen im Programmplan werden in der Programmwoche

übernommen und zusätzlich wird für dieses Objekt ein Eintrag in eine Änderungstabelle erstellt.

Folgende Änderungen können automatisch vom System erkannt werden:

Art der Änderung	Beschreibung	Änderungsname
Austausch einer Sendung	Der Film "Meine grüne Freiheit" wurde für "Der Kuss meiner Schwester" ersetzt.	Programmänderung
Startzeit ändert sich	Der Film "Metro" beginnt um 20:15 Uhr statt wie vorher geplant war um 20:00 Uhr	Zeitverschiebung
Neuer Sendetitel	Die Serie "Die Aufrechten – Aus den Akten der Strasse" heißt jetzt nur noch "Die Aufrechten"	Titeländerung
Andere Episode bei einer Serie	Die Osterfolge von "Gute Zeiten, schlechte Zeiten" wird durch die Weihnachtsfolge ersetzt.	Episodenänderung
Eine neue Sendung zusätzlich	Eine Folge von "Unter Uns" wird zusätzlich im Programmplan aufgenommen.	Zusatz
Sendung entfällt	Der Sonderablauf "Extra" wurde ersatzlos gestrichen	Entfällt

Tabelle 1 Übersicht der erkennbaren Änderungen

Alle Änderungen werden zunächst in der Änderungstabelle gesammelt. Der Redakteur kann alle Einträge ansehen und entscheiden, ob diese relevant ist oder gelöscht werden kann. Mehrere Einträge können zu einer redaktionellen Änderung zusammengefaßt und mit einer Bemerkung versehen werden, die es den Leser einfacher macht den Zusammenhang zu verstehen. So wäre es zum Beispiel sinnvoll, wenn ein Film ausgetauscht wird und sich dadurch die Startzeiten der nachfolgenden Sendungen verschieben, diese einzelnen Änderungen zu einer redaktionellen zusammenzufassen. In der Ansicht des Programmablaufplans wird ein Link zu jedem Objekt, zu dem es eine Änderung gibt, vorhanden sein. Dieser verweist immer auf den kompletten Kontext.

Alle redaktionellen Änderungen zwischen zwei Versionen werden in einer Änderungsmeldung - sortiert nach Sendetagen - zusammengefaßt. Diese Meldung wird den Verlagen auch als XML-Version zur Verfügung gestellt.

5.1.3 Personen

Eine Anforderung der Abteilung Presse/Programminformation ist eine Personendatenbank. Diese soll alle Personen enthalten, die in Filmen oder Serien mitgespielt oder im Stab gearbeitet haben.

PROFIS verwaltet diese Informationen sind aber für die Programmwochendatenbank nicht brauchbar. Zum Beispiel wird für jedes Bestandsobjekt jeder Schauspieler neu angelegt. So wird "Arnold Schwarzenegger" in "Hercules" und "Terminator 2" in der Datenbank für jedes Bestandsobjekt einzeln gespeichert. Zudem werden keine weiteren Informationen außer dem Vornamen und Nachnamen einer Person gepflegt.

Ziel ist ein einheitlicher Datenpool in normalisierter Form, in dem zu den einzelnen Personen zusätzliche Informationen abgelegt werden können. Durch Rückgabe an das Basissystem sollen die Daten wiederverwendbar sein.

Folgende Informationen sollen für die Personendatenbank zur Verfügung gestellt werden:

- Vorname
- Nachname
- Geburtstag
- Todestag
- Nationalität
- Biographie (Lebenslauf)
- Filmographie (in welchen Filme spielte die Person schon mit)
- Bemerkung (verheiratet mit; Anekdoten zur Person)
- Telefon
- Telefax
- E-Mail
- Homepage
- Star-Attribut

Über zusätzliche Tabellen für Darsteller und Stab können dann die einzelnen Personen einem Bestandsobjekt zugeordnet werden. Bei Darstellern wird der Rollenname mit angegeben und beim Stab die Funktion (Regie, Musik, Drehbuch usw.).

Das Star-Attribut entscheidet später die Darstellung innerhalb der Programmwoche. Die Namen der Stars werden fett unterlegt und die gespeicherte Informationen zur Person angezeigt.

Alle Personen werden auf der PROFIS-Datenbank zentral verwaltet, ebenso wie alle Zuordnungen und Informationen zu den Darstellern und Stäben. In der Programmwoche werden die Personen und Zuordnungen nur übernommen, wenn diese zu einer Sendung in der Programmablaufpläne gehören. Deshalb werden die Tabellen, die für die Programmwoche entworfen werden, in der PROFIS-Datenbank übernommen.

Um einen einheitlichen Datenpool zu gewährleisten, werden alle Änderungen der Attribute einer Person oder der Zuordnung zu einem Bestandsobjekt von der Programmwoche an die PROFIS-Datenbank weitergeben.

5.1.4 Bildwoche

Ziel ist es, die Formate Text und Bild miteinander zu verknüpfen.

Der Kunde soll zu einem ausgewählten Eintrag in der Programmwoche nicht nur wie bisher die Textinformationen erhalten, sondern auch das dazugehörige Bildmaterial sehen können.

Die Integration der Bilder soll weitestgehend automatisiert werden.

Es wird auch möglich sein, Material (Bilder, Word-Dokumente usw.) manuell einem Objekt der Programmwoche zuzuordnen oder falsche gesetzte Verbindungen zu lösen.

In erster Linie aber sollen Bilder möglichst über Importieren eines Textfiles, das von der Bildwoche generiert wird, mit den Sendungen in Verbindung gebracht werden.

Dieses Textfile wird von Picturedesk generiert und liegt im ASC-Format vor. Zu finden ist dieses File auf dem Fileserver der Bildwoche im Verzeichnis **btxt** und hat den Namen kwxx.asc (xx für die Kalenderwoche).

Beispiel:

```
/btxt/kw36.asc
```

Diese Datei beinhaltet die zu allen Bildern einer Woche dazugehörigen Textinformationen.

Diese Informationen liegen im folgenden Format vor:

Die Datensätze zu den einzelnen Bildern sind zeilenweise in der Datei gespeichert.

```
Datensatz 1 <CR><LF>
Datensatz 2 <CR><LF>
...
Datensatz n <CR><LF>
```

Jeder Datensatz beinhaltet folgende Felder:

- Film-Nummer bzw. Dateiname des Bildes
- Sendedatum
- Startzeit
- Programmwoche
- Nachtrag zur Programmwoche
- Sendetitel
- Originaltitel
- Episodentitel

- Genre
- Produktionsland
- Produktionsjahr
- Bildunterschrift bzw. Information
- Verteiler

Die Feldinhalte werden als Strings angegeben und durch Komma getrennt. Ein Datensatz sieht dann zum Beispiel so aus:

```
"D5165/6","03.09.2000","12.00","36. KW","","Insel am Ende der Welt","The
Island at the Top of World","","Abenteuerfilm","USA","1974","Auf der Flucht
vor Ungeheuern im Eisse, versuchen sich John Ivarsson, Oomiak, Sir Anthony
Ross und Freyja (David Hatman, Mako, Donald Sinden, Agneta Eckemyr, v. li.)
auf eine Eisscholle zu retten.,""A"
```

Die Bilder werden auf demselben Server im Verzeichnis **kwxx** (xx für die Kalenderwoche) - untergliedert nach ihrer Qualität - in den Ordnern **Thumb**, **Layout** und **Repro** abgelegt.

```
/kw36/thumb/D5165_6.jpg
/kw36/layout/D5165_6.jpg
/kw36/repro/D5165_6.jpg
```

Über ein Skript (z.B. AWK, ASP) wird später die ASC-Datei ausgelesen und die benötigten Informationen in einer Materialtabelle auf der Datenbank gespeichert. Über einen Trigger auf dieser Tabelle wird versucht, anhand des Sendetages und der Startzeit ein passendes Objekt innerhalb der Programmwoche zu finden und eine Zuordnung herzustellen.

5.2 Geplanter Vorgang zur Erstellung der Programmwoche

5.2.1 Wegfindung

Im Rahmen des Projektes wurden sich verschiedene Alternativen überlegt und auch ausgetestet. Im folgenden werden diese mit ihren Vor- und Nachteilen vorgestellt.

5.2.1.1 Alternativbetrachtung

5.2.1.1.1 CocaCola

CocaCola ist das von SAT 1 eigenentwickelte System zur Erstellung der Programmwoche für das Internet. Die Entwicklungsdauer betrug 3 Jahre. Es setzt sich aus den Komponenten Coca (Datenbank) und Cola (graphische Benutzeroberfläche) zusammen. Es baut auf das Programmplanungssystem des Senders auf und lädt initial die Daten eines Planungsstatus in die Datenbank. Über die Benutzeroberfläche wird der Programmplan angezeigt und die

einzelnen Programmsegmente können bearbeitet werden. Nachdem alle redaktionellen Texte erstellt worden sind, werden daraus HTML-Seiten generiert und publiziert.

Änderungen des Programmablaufes müssen nach Absprache mit der Fachabteilung manuell im Client nachvollzogen werden. Zeitgleich wird dafür eine Änderungsmeldung generiert.

Um aber dieses System bei RTL einsetzen zu können, wären Anpassungen der Datenbank-Basis und somit auch der Programme TRAFFIC und PROFIS notwendig, was zeit- und kostenintensiv ist. Auch wird dieses System nicht den Anforderungen der neuen Programmwoche bei RTL gerecht. Die erstellten Texte werden nicht auf der Datenbank archiviert, sondern müssen immer wieder neu geschrieben werden. Es gibt keine Möglichkeit, den Programmablaufplan auf Datenbankbasis zu aktualisieren und somit auch automatisch die Änderungsmeldungen zu generieren.

Wegen dieser fehlenden Eigenschaften und der zu hohen Kosten wurde der Entschluß gefaßt, ein eigenes System zu erstellen.

5.2.1.1.2 InfoOffice

Eine Überlegung im Rahmen der Diplomarbeit war das Redaktionssystem InfoOffice bei der Erstellung und Veröffentlichung der HTML-Seiten der Programmwoche zu benutzen. Es wird bei RTL zur Erstellung des Intra- und Internet-Auftritt genutzt und hätte deshalb nicht extra angeschafft werden müssen.

InfoOffice wurde 1996 von der Firma InfoOffice GmbH entwickelt und unterstützt den Redakteur oder Mitarbeiter auf einfache Weise beim Editieren von Websites. Es basiert auf der Verwendung von Templates, in denen der HTML-Code zur Gestaltung der Seiten mit Platzhaltern für - z.B. Links, Texte und Bilder - ausgestattet wird. Durch die Templates muß der Bearbeiter keine Kenntnisse zum Erstellen von HTML-Seiten haben. Er arbeitet innerhalb einer Vorschau, in der eine Vorlage eingebunden wird und füllt lediglich die Platzhalter mit Informationen.

InfoOffice bietet auch die Möglichkeit, die Platzhalter über eine ODBC-Verbindung direkt mit Feldern einer Datenbank-Tabelle zu koppeln. Dafür muß die Tabelle angegeben werden. Über Variablen können die Suchkriterien eingeschränkt werden.

Der Redakteur kann sich die Inhalte in der Vorschau ansehen und editieren. Alle Änderungen werden auch in der Datenbank aktualisiert.

Eine Einschränkung dieser Funktion von InfoOffice ist, daß in den Templates keine Joins über mehrere Tabellen unterstützt werden. Es ist möglich, dieses zu umgehen, da die SQL-Statements, die zur Datenbeschaffung nötig sind, auch manuell erstellt werden können. Dafür

müßte für jeden Sendetag ein eigenes Template erstellt werden und der Redakteur zudem auch noch SQL Kenntnisse besitzen. Die andere Möglichkeit wäre, die komplette Programmwoche innerhalb einer Tabelle zu realisieren. Dieses ist aber aufgrund der Komplexität nicht ohne weiteres möglich.

Ein weiteres Problem ist, daß Links mit Datum zu Integerwerten konvertiert werden. Dadurch wird bei der Auswahl eines Datums die falsche Information an die Datenbank geschickt. Es tritt deshalb ein Fehler auf. Dies ist zum Beispiel der Fall, wenn sich der Benutzer für einen bestimmten Sendetag den Kurzablaufplan ansehen will. Die einzelnen Sendetage sieht er im normalen Datumsformat. Nach Auswahl eines Tages schickt InfoOffice eine Abfrage an die Datenbank, um die Daten des Kurzablaufplans zu holen. Das Datum ist zu einen Integerwert konvertiert und wird daher von der Datenbank nicht mehr erkannt. Dies kann nur umgangen werden, wenn für jeden Sendetag eine statische Abfrage mit dem richtigen Datumsformat erstellt wird. Dafür müßten die Redakteure ebenfalls wieder SQL-Kenntnisse besitzen.

Wegen der mangelhaften Datenbankunterstützung wird InfoOffice nicht zur Erstellung und Veröffentlichung der HTML-Seiten für die Programmwoche eingesetzt, sondern dafür auch ein eigener Client entwickelt.

5.2.1.2 Meetings mit der Abteilung

Nach Betrachtung der alternativen Möglichkeiten wurde sich für eine komplett eigene Lösung für die Programmwoche entschieden. Dafür wurde eng mit der Abteilung Presse/Programminformation zusammengearbeitet. In regelmäßigen Abständen trafen sich die Redakteure und der Entwicklungsabteilung, um über die Anforderungen der neuen Programmwoche zu sprechen. Die Ergebnisse wurden in Konzepten zusammengefaßt und in einem Prototypen realisiert. Dieser Prototyp veranschaulichte die Funktionalität der Datenbank und gab den Redakteuren einen Einblick, wie später der Ablauf der Bearbeitung sein wird.

Dadurch, daß die Mitarbeiter der Programmwoche sich dabei ein genaues Bild vom neuen System machen konnten, konnten sie auch ihre Wünsche präzise äußern und Vorschläge machen. Die Anregungen und Verbesserungen wurden dann bis zum nächsten Meeting in den Prototypen aufgenommen und dann wiederum den Anwendern vorgestellt.

Der Prototyp ist nicht der richtige Client, mit denen später die Redakteure arbeiten werden, sondern dient lediglich als Muster für ein Programm, das nach den Regeln des Software-Designs entwickelt wird.

5.2.2 Beschreibung des Ablaufs zur Erstellung der neuen Programmwoche

Im folgenden wird der Ablauf beschrieben, wie in Zukunft die Programmwoche erstellt werden soll.

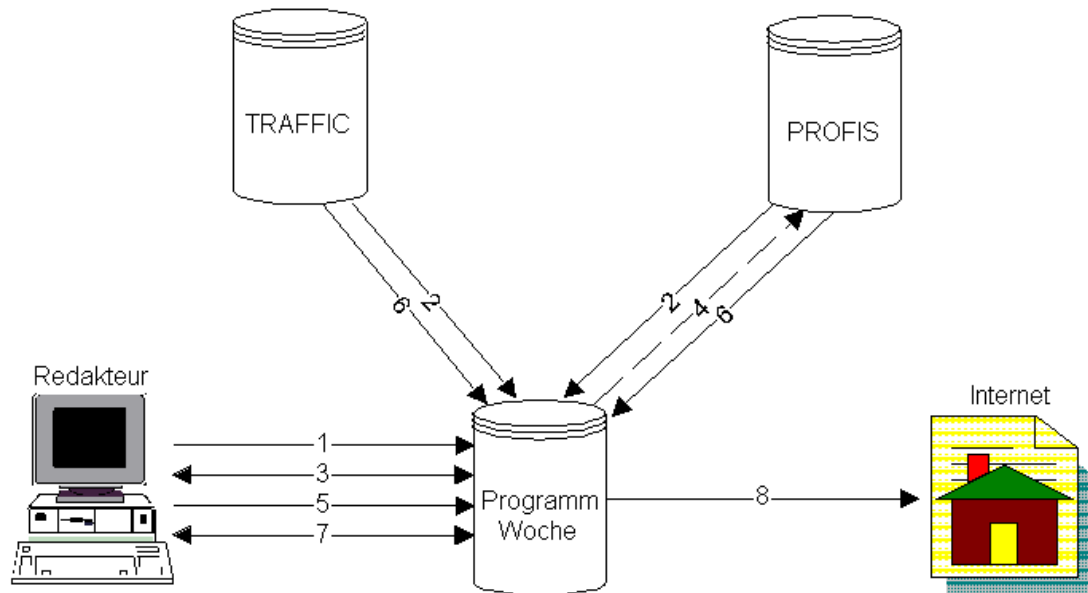


Abbildung 9 Ablauf zur Erstellung der neuen Programmwoche

1. Der Redakteur leitet über den Client initial die Füllung der Programmwochendatenbank mit den Daten der PW +7 ein.
2. Die benötigten Informationen werden aus den Systemen PROFIS und TRAFFIC gesammelt und in die Programmwochendatenbank gespeichert.
3. Die Redakteure bearbeiten die Programmwoche über den Client. Vorhandene Texte zu Sendungen und Darstellern werden gesichtet und ggf. überarbeitet bzw. neu geschrieben.
4. Änderungen der redaktionellen Daten werden von der Programmwochendatenbank an die PROFIS-Datenbank weitergegeben.
5. Starten des Abgleichs für Änderungen
6. Der vorhandene Stand der Programmwochendatenbank und der aktuell eingeplante Stand TRAFFIC – und PROFIS-Datenbank werden miteinander verglichen und eventuelle Änderungen festgehalten. Die Programmwochendatenbank wird auf den neusten Stand aktualisiert.
7. Bearbeiten der Änderungen durch den Redakteur. Einteilung in relevante und nicht relevante Änderungen. Gruppieren der relevanten Änderung aufgrund eines inhaltlichen Zusammenhangs zu redaktionellen Änderungen.

8. Veröffentlichen der aktualisierten Programmwochen und einer Änderungsmeldung, die alle redaktionellen Änderungen zwischen zwei Veröffentlichungen beinhaltet.

6 Benutzte Werkzeuge

6.1 DataArchitect

6.1.1 Einführung

Der DataArchitect stammt von der Firma Powersoft. Er ist Teil des Programmpaketes PowerDesigner, das insgesamt aus 6 verschiedenen Programmen besteht:

- ProcessAnalyst: zur Datenrettung
- AppModeler: für die physikalische Datenbankmodellierung und für die Generierung von Objekten für Applikationen
- MetaWorks: für fortgeschrittenes Teamwork, Informationsaustausch und Datenmodellmanagement
- WarehouseArchitect: für Data-Warehouse-Design und die Implementierung
- Viewer: für die Datenansicht und graphische Aufbereitung für Datenmodellinformationen
- DataArchitect: darauf wird im folgenden näher eingegangen

Der DataArchitect unterstützt den Benutzer bei der Modellierung eines Datenbankmodells, einschließlich Datenbankdesign, Generierung, reverse-engineering und Dokumentation. Er bietet einen visuellen Designer für die Erstellung eines konzeptuellen Datenbankmodell und ermöglicht ein automatisches Generieren des physikalischen Datenbankmodells (für mehrere verschiedene Datenbanksysteme). Mit der Fähigkeit des reverse-engineering können Blaupausen von Datenbanken für Dokumentations- und Unterstützungszwecke erstellt oder zu einem anderen Datenbanksystem konvertiert werden.

In diesem Projekt wurde das Tool verwendet, um das Datenbankmodell zu modellieren und eine graphische, anschauliche Vorlage zu haben. Zudem wurde anhand dieses Modells das entsprechende Skript generiert, um die Programmwochendatenbank zu erstellen. [19]

6.1.2 Anwendungsbeispiel

Um die Fähigkeiten dieses Produktes zu zeigen, wird in diesen Beispiel eine Adresdatenbank erstellt.

Beim Start des DataArchitect erhält der Benutzer folgenden Bildschirm:

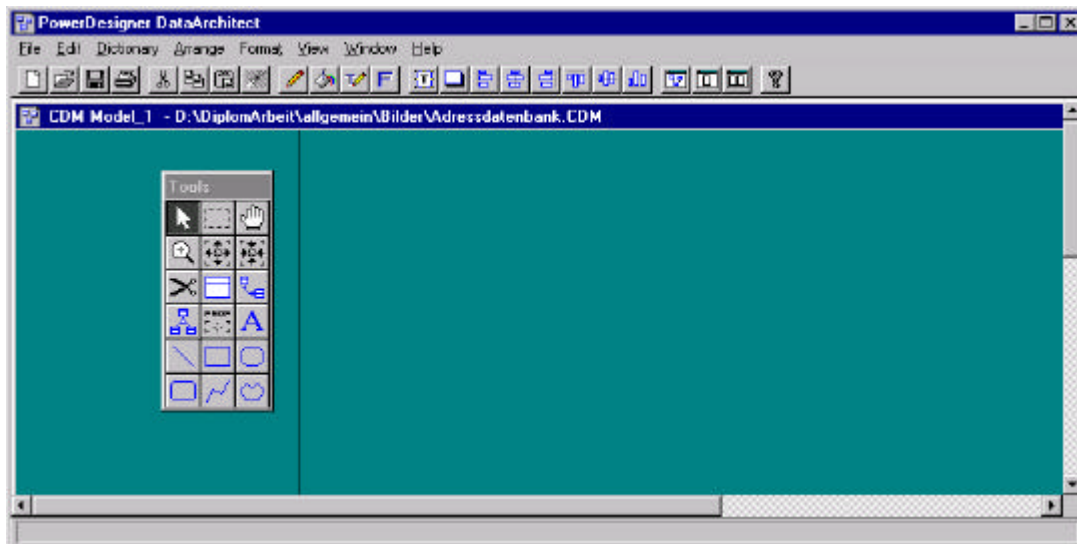


Abbildung 10 Startbildschirm

In dieser Ansicht wird das konzeptionelle Modell bearbeitet. Auf der großen Arbeitsfläche wird die spätere Datenbank modelliert. Über die Toolbar können verschiedene Elemente, die zur Darstellung notwendig sind, ausgewählt und auf der Arbeitsfläche ausgeführt werden. Als erstes müssen die Tabellen angelegt werden. Dafür wird das entsprechende Symbol der Toolbar ausgewählt und die Tabellen auf der Arbeitsfläche abgelegt.

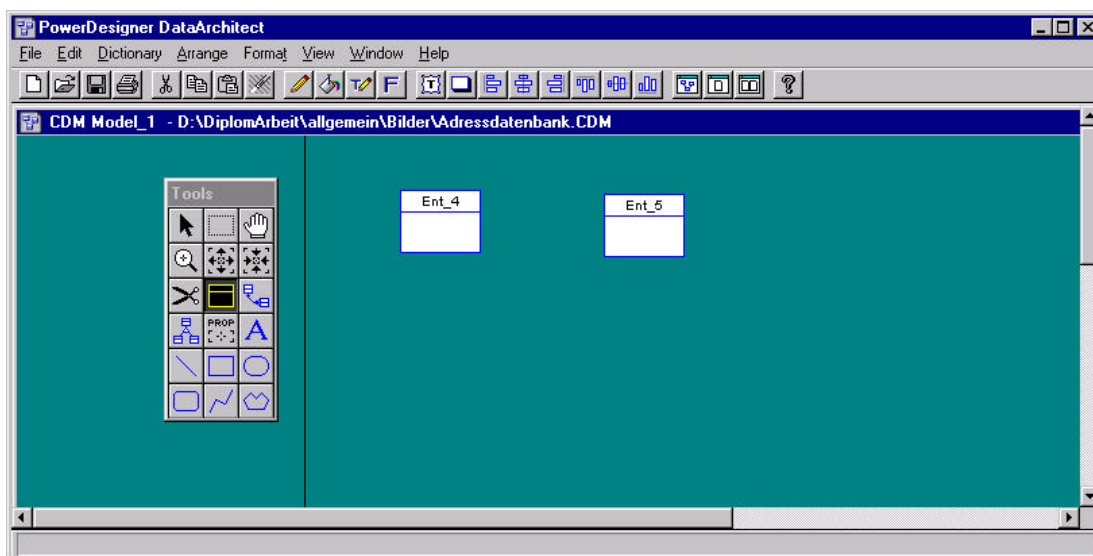


Abbildung 11 Tabellen anlegen

In den Eigenschaften der Tabelle wird zunächst der Tabellename und über die Attribute die einzelnen Felder definiert. Dabei wird auch direkt bestimmt, welche Felder die Primärschlüssel sind.

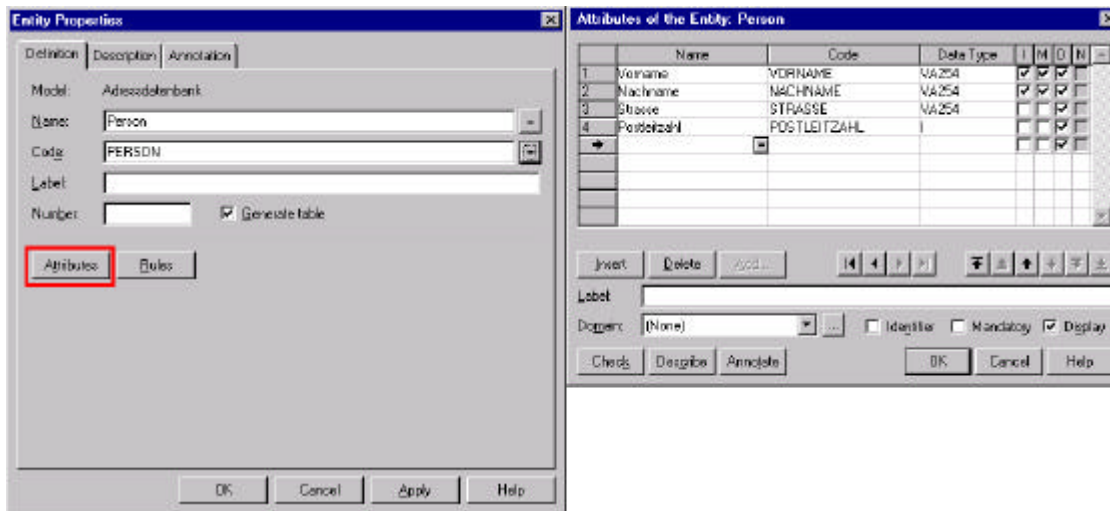


Abbildung 12 Felder in Tabellen anlegen

Wenn alle Tabellen angelegt sind, müssen die Relationen der Tabellen festgelegt werden. Dafür wird das entsprechende Symbol in der Toolbar ausgewählt und die zusammengehörigen Tabellen auf der Arbeitsfläche miteinander verbunden.

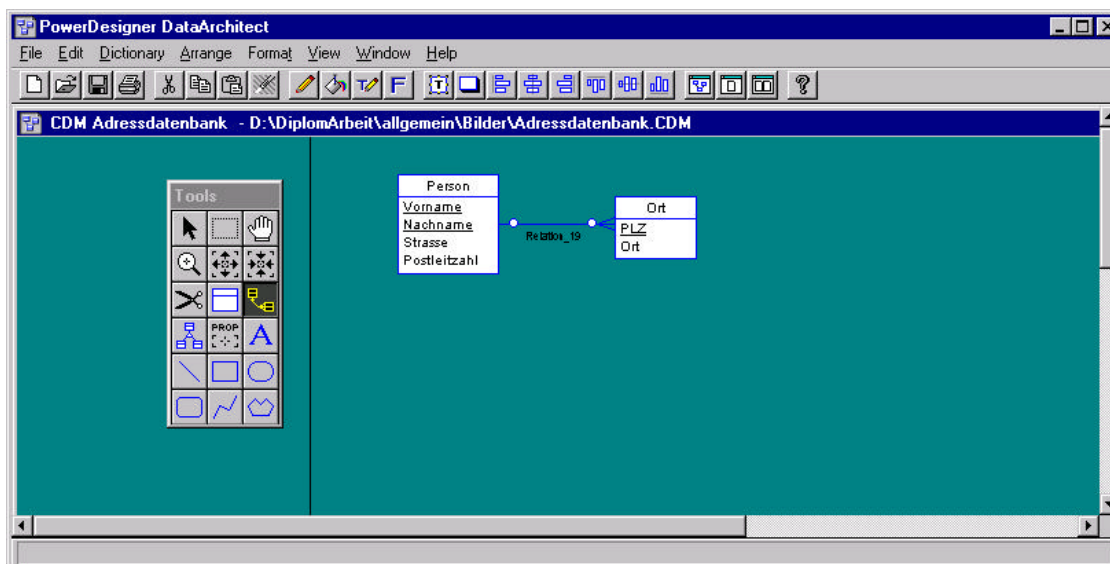


Abbildung 13 Relationen festlegen

In den Eigenschaften der Relation kann jetzt die richtige Kardinalität zwischen den beiden Tabellen angegeben werden.

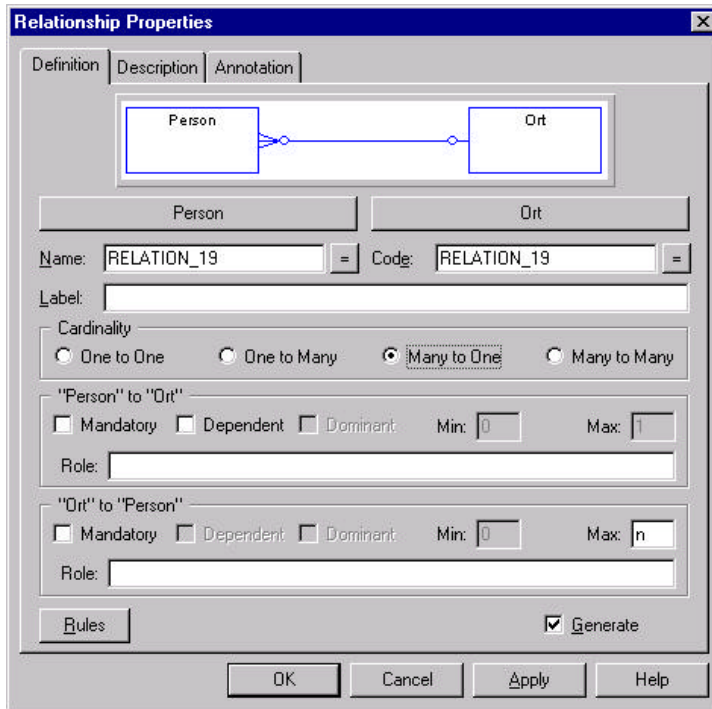


Abbildung 14 Kardinalität festlegen

Das konzeptionelle Modell ist jetzt fertiggestellt und kann in das physikalische Modell übernommen werden. Dafür muß der Menüpunkt "Generate Physical Model..." angewählt werden.

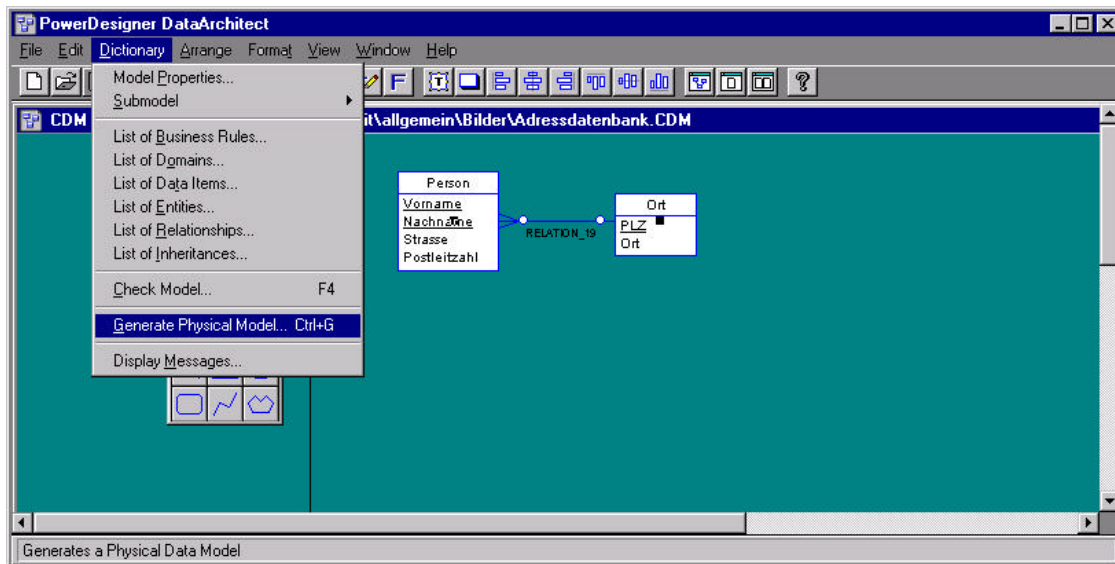


Abbildung 15 Konvertieren des Modells

Danach wird das Datenbanksystem ausgewählt, in das der Entwurf generiert werden soll. Für das physikalische Modell wird eine zusätzliche Datei angelegt. Deshalb müssen in diesen Schritt auch Pfad und Name der Datei eingegeben werden.

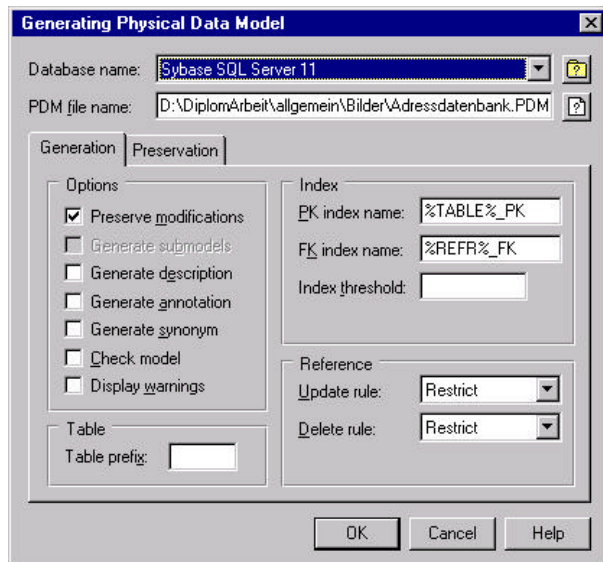


Abbildung 16 Auswahlmönü für die Generierung

Das Datenbankmodell liegt nun in der physikalischen Form vor. Es werden automatisch die Abhängigkeiten der Tabellen erzeugt. Diese müssen kontrolliert werden. Wenn alle korrekt sind, kann die Generierung der Datenbank mit Hilfe dieses Modells gestartet werden. Dies kann entweder direkt auf der Datenbank über eine ODBC-Verbindung geschehen, oder es wird ein Skript erzeugt, daß auf die Datenbank eingespielt werden kann. Bei der Generierung wird das Datenbankmodell vorher auf logische Fehler untersucht und der Benutzer darauf hingewiesen. In diesem Projekt wurde immer ein Skript generiert und die einzelnen Tabellen per Hand in die Datenbank eingetragen. Die Textdatei würde in diesem Fall folgendermaßen aussehen:

```

/*===== */
/* Database name: ADRESSDATENBANK */
/* DBMS name: Sybase SQL Server 11 */
/* Created on: 13.11.00 13:46 */
/*===== */

/*===== */
/* Table: ORT */
/*===== */
create table ORT
(
    PLZ int not null,
    ORT varchar(254) null,
    constraint PK_ORT primary key (PLZ)
)
go

```

```

/*===== */
/* Table: PERSON */
/*===== */
create table PERSON
(
    VORNAME      varchar(254)      not null,
    NACHNAME     varchar(254)      not null,
    STRASSE      varchar(254)      null   ,
    POSTLEITZAHL int               not null,
    constraint PK_PERSON primary key (VORNAME, NACHNAME)
)
go

/*===== */
/* Index: RELATION_19_FK */
/*===== */
create index RELATION_19_FK on PERSON (POSTLEITZAHL)
go

alter table PERSON
    add constraint FK_PERSON_RELATION__ORT foreign key (POSTLEITZAHL)
        references ORT (PLZ)
go

```

Dieses Skript muß jetzt nur noch auf der Datenbank eingespielt werden und die Tabellen stehen dem Anwender zu Verfügung.

6.2 Dreamweaver UltraDev

6.2.1 Einführung

Dreamweaver UltraDev stammt von der Firma Macromedia und ist ein Entwicklungstool, um HTML-Design und Clientaufbau für ASP (Active Server Pages), JSP (Sun's JavaServer Pages) und CFML (Allaire's Cold Fusion Markup Language) zu integrieren und zwar in einer visualisierten Umgebung.

Wie beim Vorgänger Dreamweaver 3.0 können HTML-Templates, CSS Styles und JavaScript benutzt werden. Zusätzlich bietet Ultradev das Einbinden von Datenbankverbindungen, das Anzeigen von serverseitigen Inhalten, das Einfügen serverseitiger Skripte und die Vorschau auf dynamische Inhalte.

Der Dreamweaver UltraDev unterstützt den Entwickler Websites an eine Datenbank anzubinden und die benötigten SQL-Statements zu generieren. Dabei kann mit jedem Datenbanksystem über ADO oder JDBC kommuniziert werden.

Er bietet server-spezifische Datenbank-Connection-Objekte an, welche es dem Benutzer ermöglichen, sich an eine Datenbank anzumelden, SQL-Abfragen durchzuführen oder gespeicherte Procedures aufzurufen. Diese Datenbank-Objekte beinhalten und unterstützen einfache und komplexe Optionen. Benutzer, die sich weniger mit SQL auskennen, können über einfache Menüs die Datenbank, die Tabelle und die Felder auswählen und sich den Inhalt auf ihrer Seite anzeigen lassen. Fortgeschrittenere Benutzer können ihre Abfragen per Hand erstellen und diese auf ihre Seite importieren. [18]

6.2.2 Anwendungsbeispiel

Dieses Beispiel zeigt, wie eine ASP-Seite erstellt wird, die dynamisch alle Inhalte der eben modellierten Adressdatenbank anzeigt.

Nach Start des Tools baut sich der Bildschirm wie folgt auf:

- Arbeitsfläche: in dieser Fläche wird das Layout der Seite definiert
- Toolbar: dort werden alle Objekte angeboten, die auf der Arbeitsfläche verwendet werden können (zum Beispiel Tabellen, Bilder usw.)
- Datenbankfunktionen: untergliedert sich in Datenbindungen (Abfragen für eine Seite erstellen und verwalten) und Serververhalten (Daten formatieren, Wiederholungsbereich anlegen).

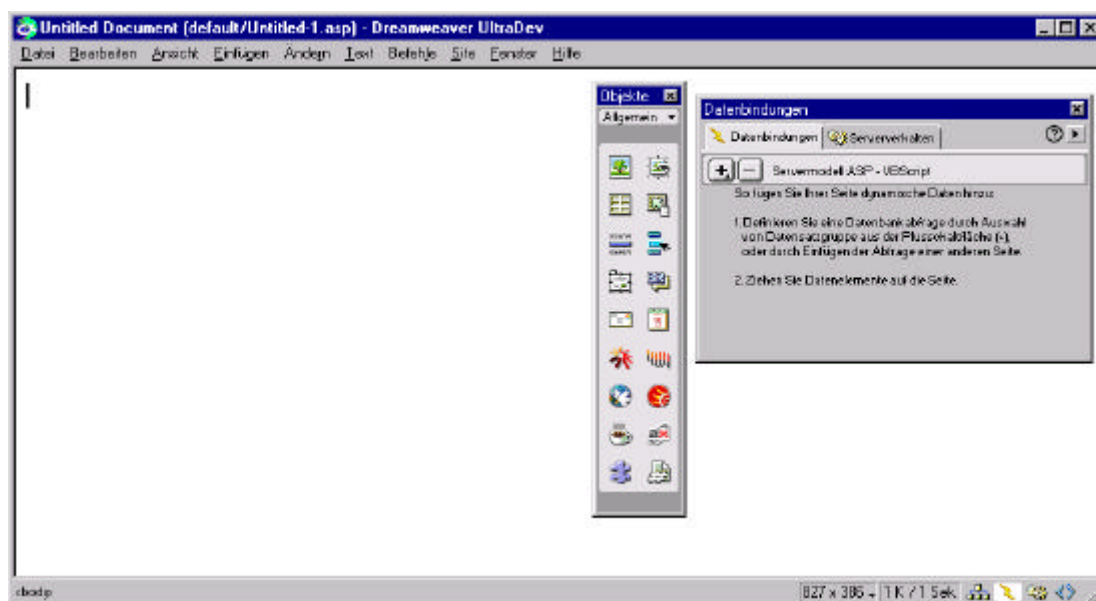


Abbildung 17 Startbildschirm UltraDev

Als erstes muß eine Datenbankverbindung über ADO hergestellt werden. Über die Menüleiste „Ändern/Verbindungen.../Neu“ wird diese Option angeboten. Der Verbindung wird ein Name gegeben, der Typ ausgewählt und der Connection-String zur Datenbank eingetragen.

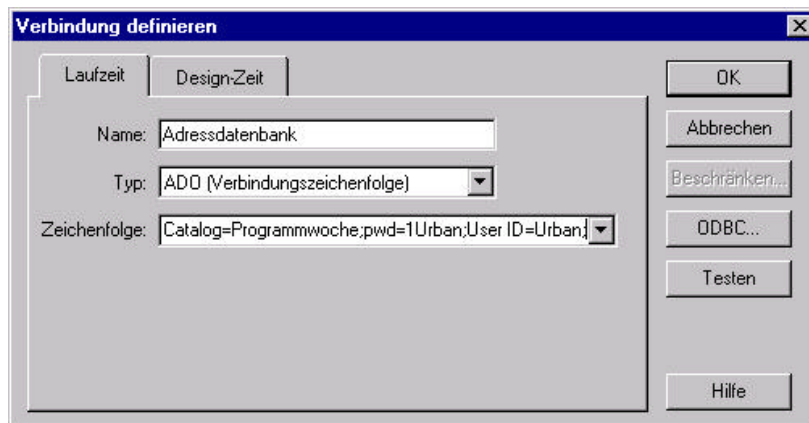


Abbildung 18 Datenquelle anlegen

Als nächstes wird über die Option 'Datenbindung' die Abfrage generiert. Über das Pluszeichen und den Unterpunkt 'Datensatzgruppe (Abfrage)' erscheint der Dialog zum Erstellen des SQL-Statements. Die Abfrage erhält einen Namen und wird der Verbindung zugeordnet. Es werden nun alle vorhanden Tabellen der Datenbank mit ihren Feldern angezeigt. Durch Auswahl der Tabellenspalten und Anklicken der Befehle wird das Statement erstellt.

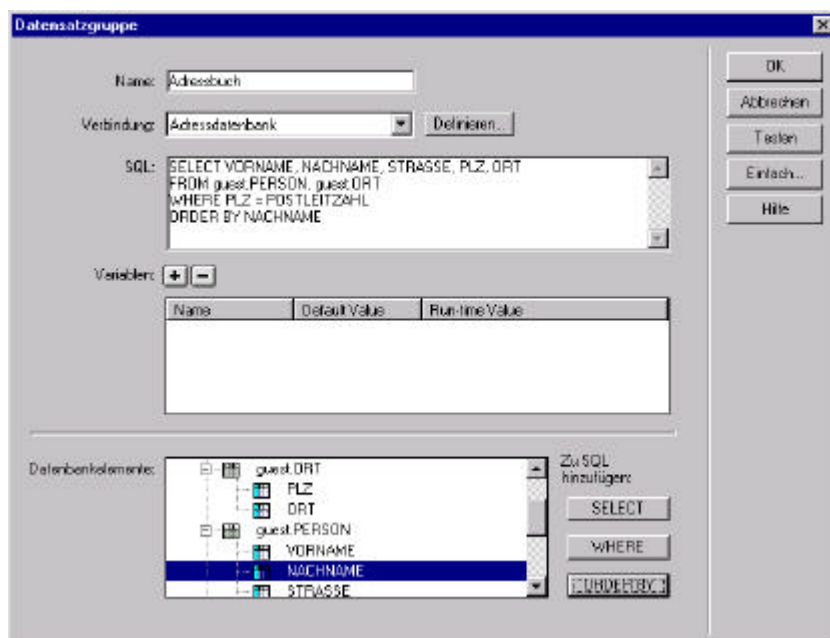


Abbildung 19 Abfrage erstellen

In der 'Datenbindung' erscheint jetzt diese Abfrage mit alle Felder, die ausgegeben werden können. Per Drag and Drop werden sie auf der Arbeitsfläche eingefügt. In diesem Beispiel

wurde über die Toolbar zuerst eine Tabelle erstellt und beschriftet, danach wurden die Felder in die Tabelle gezogen.

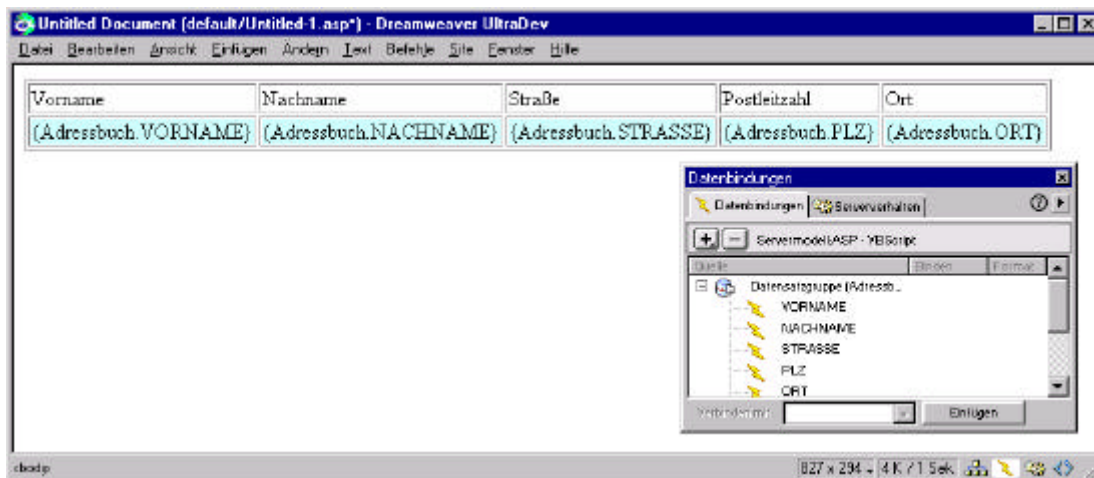


Abbildung 20 Seite erstellen

Um nicht nur den ersten Datensatz anzuzeigen, muß für diese Abfrage im Serververhalten noch eine Wiederholungsgruppe angelegt werden. Dafür werden alle Datenbankfelder markiert und die entsprechende Option im Menü ausgewählt. Im folgenden Dialog kann unterschieden werden, ob ein Teil oder alle Datensätze angezeigt werden sollen.

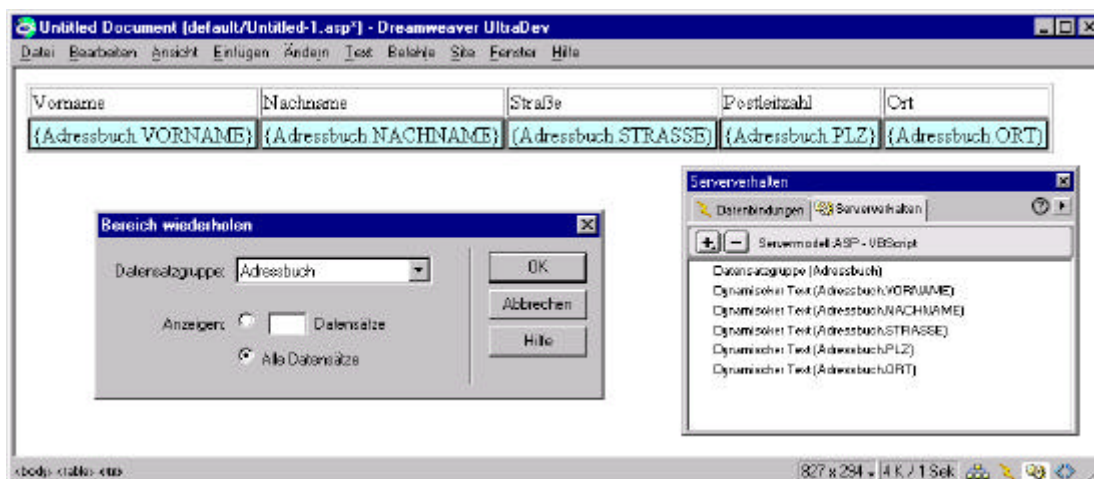


Abbildung 21 Serververhalten festlegen

Die ASP-Seite ist jetzt fertiggestellt und kann sich in der Vorschau mit den Informationen aus der Datenbank angesehen werden.

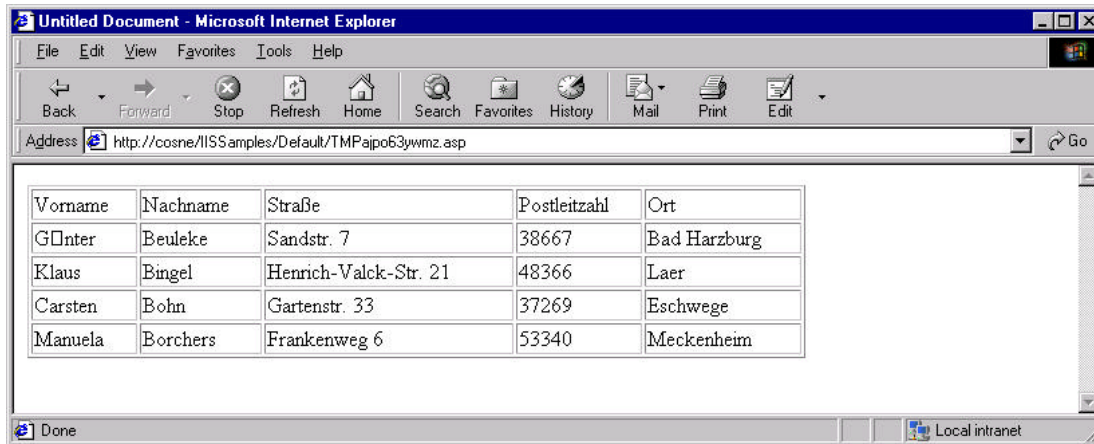


Abbildung 22 Ansicht fertige Seite

Den HTML-Code zu dieser Seite kann der Entwickler im integrierten Editor ansehen und bearbeiten. Alle Änderungen des Programmcodes werden sofort auf die Arbeitsfläche übernommen.

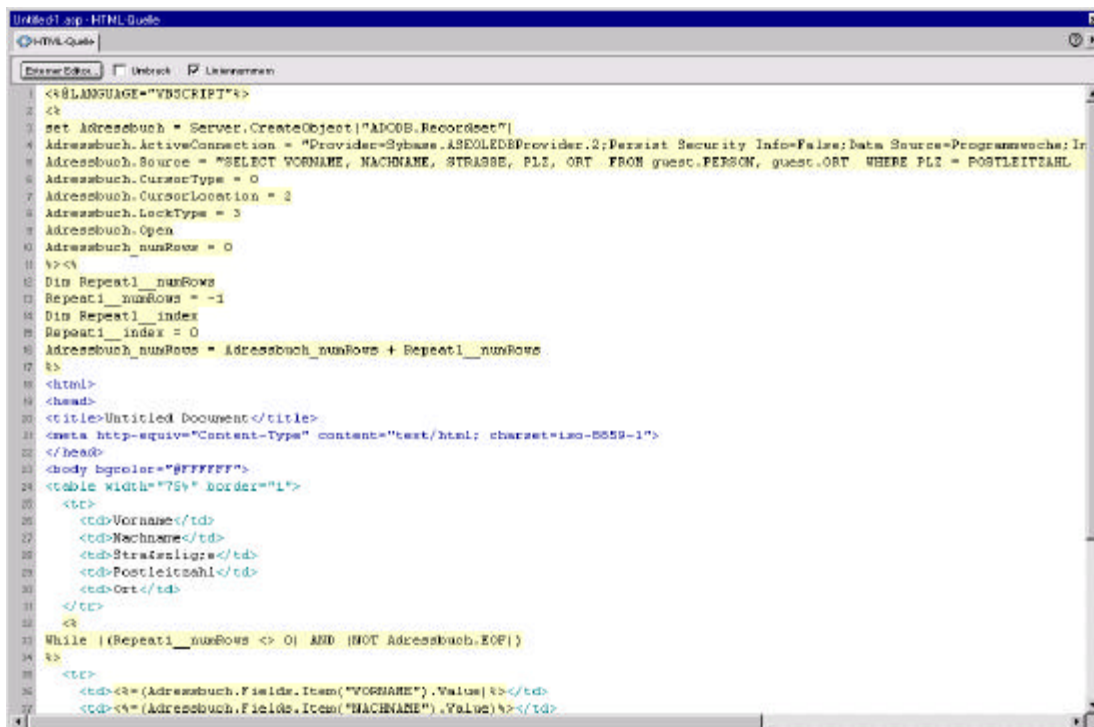


Abbildung 23 Programmcode im Editor

7 Datenmodell

7.1 Datenbankmodell der Programmwochen

Das Datenmodell richtet sich nach dem hausinternen Coding-Standard von RTL für die Programmierung von Datenbanken. Deshalb werden im Folgenden zum besseren Verständnis des Modells die Nameskonventionen des Standards erläutert:

- Tabellenbezeichner sollen nicht länger als 25 Zeichen sein
- alle Bezeichner verwenden den Singular. (Beispiel: PI_PERSON anstatt PI_PERSONEN)
- Tabellen- und Attributsbezeichner werden mit Großbuchstaben geschrieben, Wörter mit Unterstrich getrennt. (Beispiel: PI_MATERIAL_PROGRAMMWOCHEN)
- Für jeden Tabellenbezeichner wird ein projektabhängiges Präfix verwendet (<=5 Buchstaben inklusive folgendem Unterstrich). (In diesem Fall steht das Präfix PI für Programminformation)
- alle Attributsbezeichner erhalten innerhalb einer Tabelle einen Tabellenpräfix (<=5 Buchstaben inklusive folgendem Unterstrich). (Beispiel: Das Feld PW_NUMMER gehört zur Tabelle Programmwoche)
- physikalische Schlüssel werden benannt mit <Tabellenpräfix>KEY. Diese Bezeichnung wird ausschließlich für diese Schlüssel verwendet. (Beispiel: PW_KEY ist eindeutiger Schlüssel der Programmwochentabelle)
- Fremdschlüssel werden benannt mit <Tabellenpräfix><Bezeichner des Schlüssels in der Fremdtabelle>. (Beispiel: eine Änderung verweist über das Feld AEND_PW_KEY auf einen Eintrag der Programmwoche)
- die Typkürzel werden als Postfix ohne führendes „_“ geschrieben. Folgende Typkürzel wurden in diesem Projekt verwendet:

Typkürzel	Referenz	Beispiel
b	Bit	MAT_GELOESCHT_KZb
c	Char	PW_RERUN_KZc
d	Datetime	PW_SENDETAGd
i	Int	PW_KEYi
tx	Text	PW_INHALT_LANGtx
	Timestamp	TIMESTAMP
vc	Varchar	PW_INHALT_KURZvc

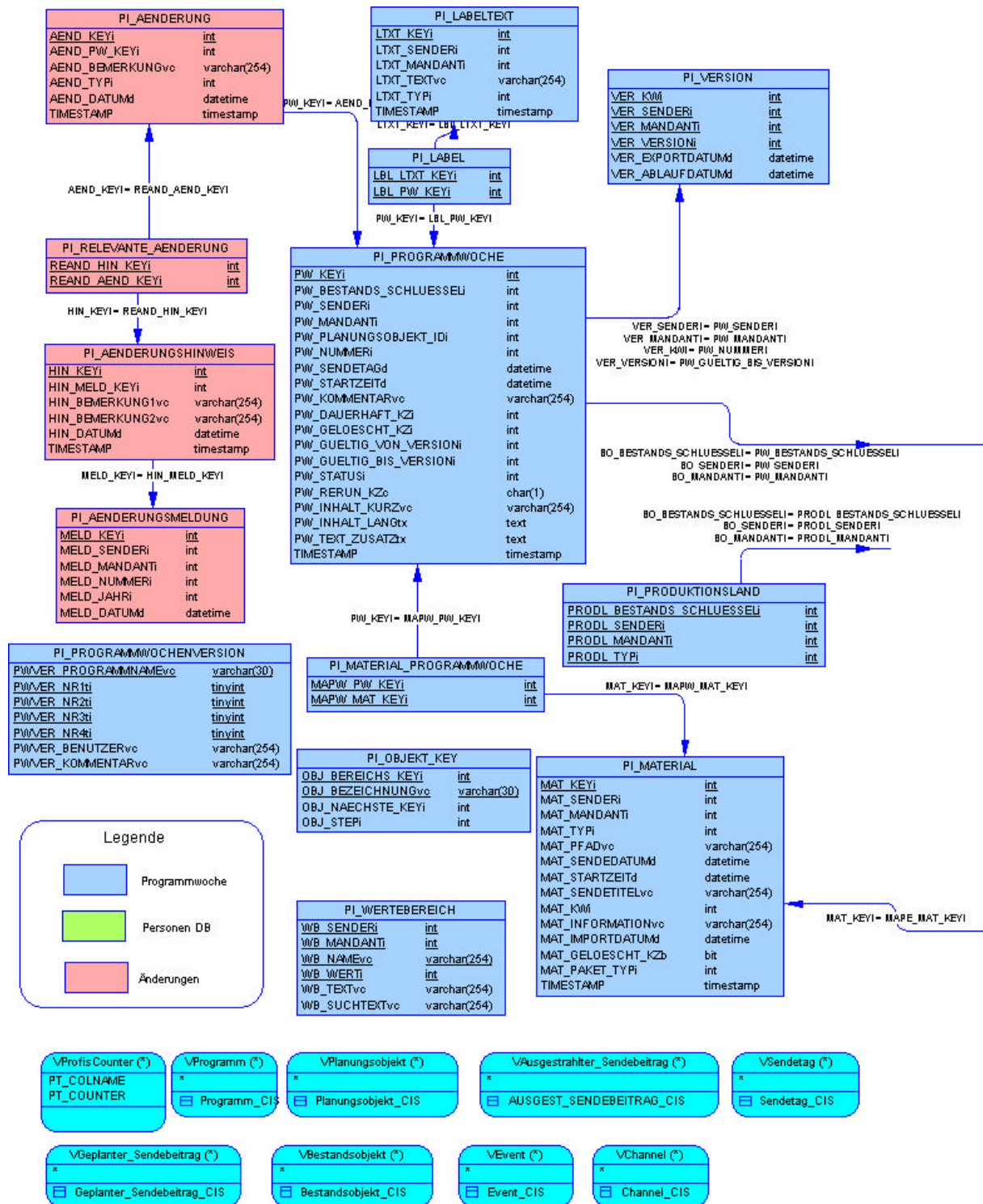


Abbildung 24 Datenbankmodell Programmwoche Teil 1

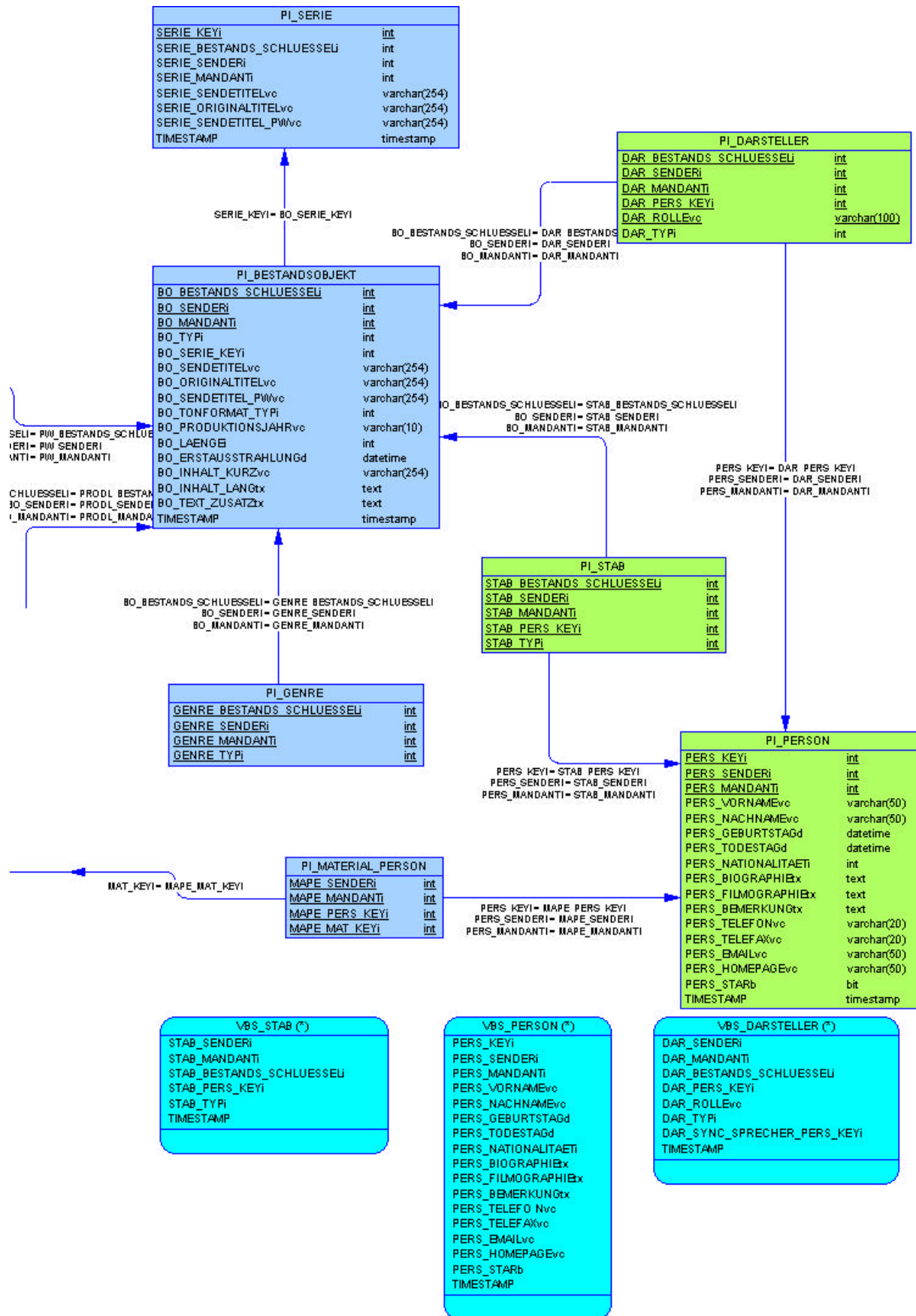


Abbildung 25 Datenbankmodell Programmwoche Teil 2

In den folgenden Abschnitten wird das zuvor dargestellte Datenbankmodell besprochen. Es erläutert die Hintergründe aller Tabellen und welche Funktion dabei die einzelnen Felder haben. Dabei wird auch auf die Trigger eingegangen, die auf den einzelnen Tabellen liegen. Zusätzlich befindet sich im Anhang unter Kapitel 11.7 eine komplette Auflistung aller Trigger, die für diese Datenbank erstellt worden sind.

7.2 Programmwoche

7.2.1 Programmwoche

Die Tabelle PI_PROGRAMMWOCHE ist das Kernstück dieser Datenbank. Hier werden zentral alle Einträge der Programmwoche verwaltet. Jeder Eintrag in dieser Tabelle erhält einen eindeutigen Programmwochenschlüssel.

Über den Bestandsschlüssel - zusammen mit der Sender- und Mandanten-ID - verweist der Eintrag der Programmwoche auf ein Element des Bestandsobjektes.

Zugleich dienen die IDs zur Unterscheidung, falls mehrere Fernsehsender über diese Datenbank verwaltet werden.

Die Planungsobjekt-ID ist eine eindeutige Kennung für einen Sendepplatz aus TRAFFIC. Diese wird später benötigt, um bei dem Änderungsabgleich die Programmpläne zwischen Programmwoche und TRAFFIC vergleichen zu können.

Über Programmwochennummer, Sendetag und Startzeit wird das Zeitgerüst des Programmablaufplans verwaltet. Die Programmwochennummer ist immer über das Sendedatum zu bestimmen, wird aber trotzdem zusätzlich zu jedem Objekt der Programmwoche gespeichert. Dadurch vereinfachen sich Abfragen auf der Datenbankseite, die die Programmpläne einer kompletten Woche betreffen (Veröffentlichung oder Ausgabe einer Programmwoche). Eine Unterscheidung zwischen Sendedatum und Startzeit ist notwendig, weil ein Sendetag um ca. 6.00 Uhr beginnt und am nächsten Tag um 5:59 Uhr endet. Eine Sendung, die am 21.10.2000 um 2:00 Uhr morgens ausgestrahlt wird, hat als Sendedatum den 20.10.2000. Da sich der Sendetagstart durch den Programmablauf des Vortages nach vorne oder hinten verschieben kann, wird das Datum des Sendetages zusätzlich verwaltet. Dadurch ist die Gruppierung der einzelnen Tage gewährleistet.

Für Bestandsobjekte von Sonderabläufen stellt PROFIS nur Dummy-Objekte zur Verfügung (z.B. "Formel Eins"). Sendungsinformationen zu diesen Objekten stehen im Kommentarfeld in der TRAFFIC-Datenbank. Dieses Feld wird ebenfalls in der Programmwochentabelle gespeichert. Der Redakteur kann sich so besser orientieren, welche Thematik der Sonderablauf hat (z.B.: Rennen in Indianapolis)

Das Kennzeichen für „Gelöscht“ bestimmt, ob ein Objekt in der Programmwoche bei der Veröffentlichung angezeigt wird. Sobald es gesetzt ist, wird der Eintrag dem Benutzer im Programmablaufplan nicht mehr angezeigt. In der Datenbank ist dieses Objekt jedoch vorhanden. Erst beim Ablauf der Programmwoche werden alle Daten zu einer Programmwoche tatsächlich gelöscht.

Bei Einträgen in der Programmwochentabelle wird über das Kennzeichen "Dauerhaft" zwischen drei Kategorien unterschieden. Beim ersten Einlesen der Daten erhalten alle Objekte der Programmwoche den Status 'initial'. Dies bedeutet, daß bei einem Änderungsabgleich dieses Objekt berücksichtigt wird. Manuell eingefügte Einträge in die Programmwoche erhalten in diesem Feld das Kennzeichen 'permanent' oder 'flüchtig'. Flüchtige Einträge werden erstellt, wenn die Basissysteme ausfallen und trotzdem eine Programmwoche veröffentlicht werden muß. Diese Objekte erscheinen bei der Veröffentlichung im Programmplan, werden jedoch beim nächsten Änderungsabgleich - ohne Erstellung eines Änderungseintrages - aus der Programmwoche gelöscht. Bei permanente Einträge dürfen die Attribute eines Objektes nicht überschrieben werden. Auch wenn im Basissystem eine Änderung zu dieser Sendung eingeplant worden ist. Ein Beispiel für die Notwendigkeit dieses Attributes ist der Programmwocheneintrag "RTL Newsflash". Diese Sendung wird in TRAFFIC eingeplant, aber nicht im Programmplan der Programmwoche mit ausgegeben. Dafür wird für dieses Objekt das Dauerhaft-Kennzeichen gesetzt und dann gelöscht. Die Sendung wird im Programmablaufplan auch nach dem nächsten Änderungsabgleich nicht mehr angezeigt.

Jedes Objekt der Programmwoche hat einen Gültigkeitszeitraum. Einträge sind ab einer bestimmten Version gültig und können bei Änderungen auch wieder ablaufen. Nähere Erläuterungen zum Thema "Versionierung" werden im Kapitel 7.1.5 beschrieben.

Über das Attribut „Status“ wird der Workflow zur Bearbeitung abgebildet. Neu angelegte Objekte der Programmwoche haben immer den Status "zu bearbeiten". Sobald ein Redakteur Änderungen gespeichert hat wird der Status des Eintrages auf "bearbeitet" gesetzt. Um in den letzten Status ("veröffentlicht") zu kommen, muß der Bearbeiter vorher jedes Objekt kontrollieren und auch auf diesen Status setzen.

Die redaktionellen Texte werden beim Anlegen eines Programmwocheneintrages aus der Bestandsobjekttabelle übernommen. Dadurch werden die redaktionellen Texte doppelt verwaltet, nämlich in der Programmwoche und im Bestandsobjekt. Aber aus zwei Gründe ist es notwendig, diese Felder in der Programmwochentabelle zu halten. Zum einen gibt es in PROFIS keine richtigen Bestandsobjekte zu kurzfristigem Sendematerial (Sonderabläufe),

sondern lediglich Dummy-Einträge. Es gibt z.B. nur ein Objekt für 'Formel Eins'. Wenn alle zwei Wochen ein Rennen stattfindet, verweisen 3 Einträge der Programmwoche auf diesen Sonderablauf in der Tabelle „Bestandsobjekt“. Der Redakteur will für jedes einzelne Rennen in Indianapolis, Suzuka und Malaysia individuelle Erläuterungen schreiben. Dieses ist nicht zu realisieren, wenn die redaktionellen Texte am Bestandsobjekt festgemacht sind. Das andere Problem sind Reruns, die in dieser Tabelle über das Rerun-Kennzeichen gepflegt werden. Reruns sind Wiederholungen eines Filmes oder einer Episode innerhalb von 24 Stunden. Hintergrund dafür ist, daß diese Wiederholungen einen Sender keine zusätzliche Lizenzgebühr kosten. In der Programmwoche soll es aber möglich sein, für einen Rerun einen individuellen Text zu erstellen.

Alle Änderungen in den redaktionellen Daten werden über einen Trigger an das ursprüngliche Bestandsobjekt weitergegeben, wenn es sich nicht um einen Sonderablauf oder einen Rerun handelt.

Der TIMESTAMP sichert, daß sich zwei am gleichen Objekt arbeitende Redakteure, ihre Änderungen nicht gegenseitig überschreiben.

PI PROGRAMMWOCH

Feld	Beschreibung
PW_KEYi	eindeutiger Schlüssel eines Objektes der Programmwoche
PW_BESTANDS_SCHLUESSELi	eindeutige Zuordnung zu einem Bestandsobjekt, zusammen mit der Sender- und Mandanten-ID
PW_SENDERi	Zugehörigkeit des Eintrages zu einem Sender.
PW_MANDANTi	Zugehörigkeit des Eintrages zu einem Mandanten.
PW_PLANUNGSOBJEKT_IDi	eindeutige Zuordnung zu einen Sendepplatz aus TRAFFIC
PW_NUMMERi	Programmwoche der geplanten Ausstrahlung
PW_SENDETAGd	Datum der geplanten Ausstrahlung
PW_STARTZEITd	Startzeit der geplanten Ausstrahlung
PW_KOMMENTARvc	Zusatzinformationen zu Sonderabläufen
PW_DAUERHAFT_KZi	Kennzeichen, ob der Eintrag initial, dauerhaft oder flüchtig ist
PW_GELOESCHT_KZb	Kennzeichen, ob der Eintrag in der Programmwoche gelöscht ist

PW_GUELTIG_VON_VERSIONi	Version, ab dem der Eintrag gültig ist
PW_GUELTIG_BIS_VERSIONi	Version, ab dem der Eintrag ungültig ist
PW_STATUSi	Bearbeitungsstatus
PW_RERUN_KZc	Kennzeichen, ob es sich bei diesem Programmwocheneintrag um einen Rerun handelt
PW_INHALT_KURZvc	Kurzinhalt einer Sendung
PW_INHALT_LANGtx	Langinhalt einer Sendung
PW_TEXT_KURZtx	Randnotizen zu einer Sendung
TIMESTAMP	

Beim Einfügen eines Programmwochenobjektes, wird über einen Trigger geprüft, ob ein passendes Bestandsobjekt in der Tabelle PI_BESTANDSOBJEKT vorhanden ist. Gleiches passiert, wenn der Bestandsschlüssel geändert wird. Ebenso muß für einen Eintrag in der Programmwoche ein passende Version in der Versionstabelle vorhanden sein. Schlägt diese Prüfung fehl, werden alle Änderungen rückgängig gemacht. Beim tatsächlichen Löschen eines Programmwochenelements überprüft ein Delete-Trigger, ob keine Einträge in den Tabellen PI_AENDERUNG, PI_LABEL und PI_MATERIAL_PROGRAMMWOCHHE zu diesem Objekt vorhanden sind. Findet der Trigger einen Verweis in einer dieser Tabellen, verhindert dieser das Löschen.

7.2.2 Label

Label sind zusätzliche Informationen zu einem Programmplatz oder zu einer Sendung. Es wird dabei zwischen folgenden Labeltypen unterschieden:

Labeltyp	Beispiel	Beschreibung
Themenlabel	Willkommen zu Hause	gruppiert ein Thema zu einem Zeitbereich
Sendungslabel	Free TV-Premiere	kennzeichnet die Besonderheit einer Sendung
Altersempfehlung	RTL-Altersempfehlung: ab 6 Jahre	weist auf eine Altersempfehlung bei Serien oder Filmen hin.
Videotextseite	Infos zur Sendung auf RTL-Text S. 325	Wo können weitere Informationen zu einer Sendung gefunden werden?

Die verschiedenen Label werden in der Labeltexttabelle verwaltet. Dabei erhalten sie einen eindeutigen Schlüssel. Zugleich werden über die Sender- und Mandanten-ID die Labels für verschiedene Sender verwaltet. Der Text enthält die Informationen, die später zu einem Objekt der Programmwoche ausgegeben werden sollen. Über das Attribut Typ wird die Zuordnung zu einem Labeltypen hergestellt, dessen Werte in der Wertebereichstabelle definiert sind.

Der TIMESTAMP dient hier ebenfalls dazu, daß sich zwei am gleichen Objekt arbeitende Redakteure, ihre Änderungen nicht gegenseitig überschreiben können.

PI_LABELTEXT

Feld	Beschreibung
LTXT_KEYi	eindeutiger Schlüssel eines Labels
LTXT_SENDERi	Zugehörigkeit des Eintrages zu einem Sender.
LTXT_MANDANTi	Zugehörigkeit des Eintrages zu einem Mandanten.
LTXT_TEXTvc	Ausgabertext in der Programmwoche
LTXT_TYPi	Verweis auf einen Eintrag im Wertebereich für "Label"
TIMESTAMP	

Beim Löschen eines Labeltextes überprüft ein Delete-Trigger, ob es zu diesem noch einen Eintrag in der Tabelle PI_LABEL gibt. Sollte dies der Fall sein, schlägt das Löschen fehl.

Die Zuordnung der Label zu einem Programmwochenelement geschieht in der Labeltabelle. Hier wird der jeweilige Labeltext- und Programmwochenschlüssel gespeichert. Dabei kann es auch mehrere verschiedene Label für einen Eintrag in der Programmwoche geben. Umgekehrt kann ein Label auch mehren Sendungen zugewiesen werden.

PI_LABEL

Feld	Beschreibung
LBL_LTXT_KEYi	eindeutiger Schlüssel aus der Labeltexttabelle
LBL_PW_KEYi	eindeutiger Schlüssel aus der Programmwochentabelle

Beim Einfügen oder Ändern eines Eintrages in dieser Tabelle wird über Trigger überprüft, ob die Schlüssel zu einem Labeltext und einem Programmwocheneintrag in der Programmwochendatenbank vorhanden sind. Sollte eine der beiden Zuordnungen nicht zutreffen, verhindert der Trigger den Eintrag in diese Tabelle.

7.2.3 Bestandsobjekt

Bestandsobjekte sind alle Filme, Episoden und Sonderabläufe. Diese werden in der Tabelle PI_BESTANDSOBJEKT verwaltet. Jede Sendung hat ihren eindeutigen Schlüssel über die Bestandsobjektnummer, Sender- und Mandanten-ID. Dieser Schlüssel wird, sofern es sich nicht um abstrakte Einträge handelt, nicht von der Programmwochendatenbank generiert, sondern aus PROFIS übernommen. Dadurch bleibt die Verbindung zum Originalobjekt im Basissystem bestehen und es können die Änderungen der redaktionellen Texte dorthin weitergegeben werden.

Teilweise kann es vorkommen, daß beim Erstellen der Programmwoche die Planung eines Sendetages noch nicht abgeschlossen ist und Platzhalter für einen Sendeplatz im Programmplan eingefügt worden sind. Für diesen Fall werden abstrakte Bestandsobjekte in der Programmwochendatenbank erstellt. Sie erhalten einen negativen Bestandsschlüssel und sind nur in der Programmwochendatenbank vorhanden. Diese Objekte können, wie alle anderen bearbeitet werden, nur werden keine Änderungen an das Basissystem weitergegeben. Es wird ebenfalls ein Bestandsobjekt mit negativem Schlüssel angelegt, wenn es sich um eine Sendung handelt, der manuell in die Programmwoche eingefügt wurde.

Über den Typ des Bestandsobjektes wird unterschieden, ob es sich bei einem Bestandsobjekt um einen Film, eine Serie oder einen Sonderablauf handelt. Diese Angabe ist wichtig, um die Ausgabe des Langablaufplans richtig aufbauen zu können. Eine Serie enthält mehr Informationen in der Darstellung als ein Sonderablauf.

Wenn in der Bestandsobjekttabelle von Serien gesprochen wird, handelt es sich nur um einen Oberbegriff. In Wirklichkeit werden hier die Daten von Episoden zu Serien gespeichert. In der Programmwoche muß aber neben dem Episodentitel auch der Serientitel ausgegeben werden. Dafür gibt es eine spezielle Serientabelle, auf die später noch näher eingegangen wird. Um einen Verweis auf einen Eintrag dieser Tabelle zu haben, wird im Bestandsobjekt des Typs 'Serie' der dazugehörige Serienschlüssel abgelegt. Bei allen anderen Einträgen steht dort ein NULL-Wert.

Ein Bestandsobjekt besitzt drei verschiedene Sendetitel: Sendetitel, Originaltitel und Programmwochen-Sendetitel. Der Sendetitel und der Originaltitel werden aus PROFIS übernommen. Eigentlich würden diese beiden Einträge für die Programmwoche genügen. Wenn der Titel einer Sendung einen Artikel enthält (der, die, das), wird dieser an das Ende gestellt. Aus "Der Spezialist" wird in PROFIS: "Spezialist, Der". In der publizierten Programmwoche soll dieser PROFIS-Sendetitel nicht erscheinen. Beim Änderungsabgleich sollen auch Sendetiteländerungen erkannt werden. Diese können aber nur mit dem ursprünglichen Titel aus PROFIS verglichen werden. Deshalb werden beide Sendetitel in der

Tabelle gehalten. Sobald der PROFIS-Sendetitel im Bestandsobjekt eingetragen oder verändert wird, prüft ein Trigger auf dieser Tabelle den Eintrag, bereinigt ihn und speichert ihn als Programmwochen-Sendetitel ab.

Tonformat und Länge werden zwar innerhalb der Programmwoche im Bestandsobjekt gespeichert, kommen aber ursprünglich aus TRAFFIC. Es sind Informationen, die zwar zu einer Sendung gehören, aber variieren können, je nachdem welches Band für eine Ausstrahlung eingesetzt wird. Ein Film in der Fassung ab 16 Jahre ist durch die geschnittenen Szenen kürzer als die Version ab 18. Die verschiedenen Bänder werden in TRAFFIC verwaltet und eingeplant.

Für die Programmwoche sind diese Felder trotzdem dem Bestandsobjekt zugeordnet. Zwar könnten die Bänder bei Wiederholungen variieren, aber für die Programmwoche ist an dieser Stelle die Genauigkeit vernachlässigbar.

Das Produktionsjahr und der Erstausrstrahlungstermin sind Informationen, die wiederum aus PROFIS stammen. Das Produktionsjahr ist ein Textfeld und wird direkt aus PROFIS übernommen.

Für jedes Bestandsobjekt werden in PROFIS alle Ausstrahlungstermine gespeichert. Das Datum der Erstausrstrahlung ist das früheste Datum aus dieser Liste. Bei einer Premiere sind natürlich keine Ausstrahlungstermine vorhanden und es wird dementsprechend kein Datum in das Feld Erstausrstrahlung eingetragen.

Die redaktionellen Texte werden hier für jedes Bestandsobjekt zentral verwaltet. Bezieht sich ein Eintrag der Programmwoche auf ein Bestandsobjekt, werden die Inhalte weitergegeben, wenn es sich nicht um einen Rerun oder einen Sonderablauf handelt. Änderungen der redaktionellen Texte in der Programmwochentabelle werden zuerst im Bestandsobjekt gespeichert, danach über einen Trigger an PROFIS weitergegeben.

Der TIMESTAMP dient zur Sicherheit, daß zwei Redakteure, die am gleichen Objekt arbeiten, sich gegenseitig ihre Änderungen nicht überschreiben.

PI_BESTANDSOBJEKT

Feld	Beschreibung
BO_BESTANDS_SCHLUESSELi	Teil des eindeutigen Schlüssels innerhalb der Programmwoche und Verbindung zum Originalobjekt aus PROFIS
BO_SENDERi	Teil des eindeutigen Schlüssels innerhalb der Programmwoche und Verbindung zum

	Originalobjekt aus PROFIS
BO_MANDANTi	Teil des eindeutigen Schlüssels innerhalb der Programmwoche und Verbindung zum Originalobjekt aus PROFIS
BO_TYPi	Art der Sendung (Film, Serie oder Sonderablauf)
BO_SERIE_KEYi	Verweis auf einen Eintrag in PI_SERIE
BO_SENDETITELvc	Sendetitel PROFIS
BO_ORIGINALTITELvc	Originalsendetitel
BO_SENDETITEL_PWvc	Sendetitel Programmwoche
BO_TONFORMAT_TYPi	Verweis auf einen Eintrag im Wertebereich für "Vertonung"
BO_PRODUKTIONSJAHRvc	Produktionsjahr
BO_LAENGEi	Sendelänge
BO_ERSTAUSSTRAHLUNGd	Datum der Erstausstrahlung
BO_INHALT_KURZvc	Kurzinhalt einer Sendung
BO_INHALT_LANGtx	Langinhalt einer Sendung
BO_TEXT_ZUSATZtx	Randnotizen zu einer Sendung
TIMESTAMP	

Um die Datenkonsistenz zu gewährleisten, liegen auf dieser Tabelle Trigger für Insert, Update und Delete. Beim Löschen muß darauf geachtet werden, daß es keine Verbindungen mehr zu Einträgen in den Tabellen PI_PROGRAMMWOCH, PI_PRODUKTIONSLAND, PI_GENRE, PI_STAB und PI_DARSTELLER gibt. Werden Einträge dieser Art gefunden, kann das Bestandsobjekt nicht gelöscht werden. Die Bestandsobjekttabelle besitzt aber auch eine Abhängigkeit zu der Serientabelle. Der Serienschlüssel muß in der Tabelle PI_SERIE vorhanden sein. Beim Einfügen einer Episode oder Ändern dieses Schlüssels wird der Eintrag endgültig geschrieben, wenn eine entsprechende Serie dazu gefunden worden ist. Andernfalls werden alle Änderungen über einen Rollback zurückgenommen.

Neben der Funktionalität die Konsistenz der Daten sichern, übernehmen die Trigger auch die Aufgabe, Daten mit den Basissystemen auszutauschen. Eine bereits vorher beschriebene Aufgabe, ist die Aktualisierung der redaktionellen Daten im Bestandsobjekt aus dem Basissystem. Eine weitere wird schon beim Anlegen eines Objektes in dieser Tabelle ausgeführt. Gefüllt wird diese Tabelle über die Stored Procedures ‚GetAktuelleProgrammwoche‘ und ‚AenderungUeberpruefung‘. Die Procedures füllt beim

Anlegen eines Bestandsobjekts lediglich die Felder Bestandsschlüssel, Sender, Mandant, Typ, Länge und Vertonung. Alle zusätzlichen Informationen - wie Sendetitel, Produktionsjahr, Erstausstrahlung und redaktionelle Texte - werden über den Insert-Trigger hinzugefügt. Bei einer Episode wird gegebenenfalls eine Serie angelegt, wenn diese noch nicht in der Programmwochendatenbank vorhanden ist. Gibt es schon den Eintrag in der Serientabelle, wird der passende Serienschlüssel herausgesucht und zum Programmwochenbestandsobjekt gespeichert. Ebenfalls werden die Tabellen Genre und Produktionsland mit den Informationen aus PROFIS gefüllt. Weiter sorgt der Trigger dafür, daß die Personen, die zu einer Sendung gehören, angelegt werden, falls sie noch nicht in der Programmwochendatenbank vorhanden sind. Die Zuordnungen der Stäbe und Darsteller aus PROFIS zu einem Bestandsobjekt und einer Person werden ebenfalls übernommen.

Für Episoden in der Bestandsobjekttabelle werden weitere Informationen zur Serie benötigt. Diese werden in der Tabelle PI_SERIE verwaltet.

Für jede Serie innerhalb der Programmwochendatenbank wird ein eindeutiger Schlüssel generiert. Mit diesem Schlüssel verweist eine Episode auf ein Element der Serientabelle.

Über den Bestandsschlüssel, die Sender-ID und der Mandanten-ID hält ein Eintrag die Verbindung zum Originalobjekt aus PROFIS. Bei einem Änderungsabgleich können somit Unterschiede zwischen den Serien der Programmwoche und dem Original entdeckt werden.

Auch hier werden drei Sendetitel verwaltet. Wie beim Bestandsobjekt wird ein Feld zum Vergleichen, die beiden anderen zum Publizieren benötigt.

PI_SERIE

Feld	Beschreibung
SERIE_KEYi	eindeutiger Schlüssel
SERIE_BESTANDS_SCHLUESSELi	Verbindung zum Originalobjekt aus PROFIS
SERIE_SENDERi	Verbindung zum Originalobjekt aus PROFIS
SERIE_MANDANTI	Verbindung zum Originalobjekt aus PROFIS
SERIE_SENDETITELvc	Sendetitel PROFIS
SERIE_ORIGINALTITELvc	Originalsendetitel
SERIE_SENDETITEL_PWvc	Sendetitel Programmwoche
TIMESTAMP	

Aus Gründen der Datenkonsistenz wird beim Löschen aus dieser Tabelle überprüft, ob Verbindungen zu Bestandsobjekten bestehen. Nur wenn es keine Beziehungen mehr gibt, wird die Serie gelöscht.

Im Gegensatz zum Basissystem werden die Genres und Produktionsländer für jedes Bestandsobjekt in normalisierter Form in eigenen Tabellen verwaltet.

PROFIS stellt zwar die Informationen dazu zur Verfügung, benutzt aber einen String, in dem die Abkürzungen der einzelnen Einträge durch ein Backslash getrennt hintereinander stehen. (z.B.: Genre: COM/FAM oder Produktionsland: DE/FR/IT)

Beim Einfügen eines Bestandsobjektes wird über einen Trigger dieser String aus PROFIS eingelesen und an eine Stored Procedure weitergegeben. Diese verarbeitet den String, sucht sich aus dem Wertebereich den entsprechenden Typ heraus und speichert die einzelnen Elemente zu einem Bestandsobjekt in die jeweilige Tabelle.

PI_GENRE

Feld	Beschreibung
GENRE_BESTANDS_SCHLUESSELi	Teil des eindeutigen Schlüssels für einen Eintrag in PI_Bestandsobjekt
GENRE_SENDERi	Teil des eindeutigen Schlüssels für einen Eintrag in PI_Bestandsobjekt
GENRE_MANDANTi	Teil des eindeutigen Schlüssels für einen Eintrag in PI_Bestandsobjekt
GENRE_TYPi	Verweis auf einen Eintrag im Wertebereich für "Genres"

PI_PRODUKTIONS LAND

Feld	Beschreibung
PRODL_BESTANDS_SCHLUESSELi	Teil des eindeutigen Schlüssels für einen Eintrag in PI_Bestandsobjekt
PRODL_SENDERi	Teil des eindeutigen Schlüssels für einen Eintrag in PI_Bestandsobjekt
PRODL_MANDANTi	Teil des eindeutigen Schlüssels für einen Eintrag in PI_Bestandsobjekt
PRODL_TYPi	Verweis auf einen Eintrag im Wertebereich für "Land"

Auf beiden Tabellen liegt ein Insert- und Update-Trigger, der überprüft, ob das angegebene Bestandsobjekt vorhanden ist und gegebenenfalls den Eintrag wieder rückgängig macht.

7.2.4 Material

Die Materialtabelle dient als Sammelcontainer für Materialien wie Pressemappen, Bilder usw. Optimiert ist die Tabelle für den Import des ASC-Files der Abteilung Bildwoche. Es können aber auch Einträge manuell hinzugefügt werden, die nicht auf Bildmaterial verweisen.

Jeder Eintrag erhält beim Einfügen einen eindeutigen Materialschlüssel. Über die Mandanten- und Sender-ID wird unterschieden, welchem Fernsehsender das Material zugeordnet werden kann.

Über den Typen wird definiert, ob es sich um ein Bild, ein Word-Dokument usw. handelt. Dieser wird über den Wertebereich definiert, der auch die dazugehörigen Dateiendungen verwaltet.

Der Pfad enthält die Information, an welcher Stelle das Material gefunden werden kann. In diesem Feld findet sich der komplette Pfad mit Server, Unterordnern und Dateinamen.

Die Felder Sendedatum, Startzeit, Sendetitel und Programmwoche enthalten Informationen, die es ermöglichen, einen passenden Eintrag in der Programmwochentabelle finden zu können. Um später das Material in der Programmwoche anzeigen zu können, muß es eine Zuordnung zu einem Objekt der Programmwoche oder einer Person haben. Diese werden über die Tabellen `PI_MATERIAL_PEOGRAMMWOCH` und `PI_MATERIAL_PERSON` realisiert, auf die später noch näher eingegangen wird.

Im Informationsfeld stehen Bemerkungen oder Erklärungen zum eingefügten Material. Bei Bildmaterial wird dort eine Bildunterschrift eingetragen.

Im Feld Importdatum wird das Datum und die Uhrzeit festgehalten, wann ein Eintrag eingefügt worden ist. Dadurch kann kontrolliert werden, welche Bildwoche schon importiert worden ist und wann dies geschehen ist.

Einträge dieser Tabelle werden erst dann gelöscht, wenn die dazugehörige Programmwoche abgelaufen ist und aus der Datenbank gelöscht wird. Soll ein Eintrag schon vorher für den Benutzer nicht mehr sichtbar sein, so wird dafür lediglich das Gelöscht-Kennzeichen gesetzt.

Über das Paket-Attribut können Bilder in verschiedene Gruppen eingeordnet werden. Folgende Gruppen sind bisher vorgesehen und werden über den Wertebereich gepflegt:

Typ	Gruppe	Beschreibung
0	Default	-
1	Programmwoche	Das Bild wird zum dazugehörigen Programmwocheneintrag angezeigt.
2	Highlight	Das Bild wird beim Highlight der Woche angezeigt.
4	Kinderprogramm	Das Bild gehört zum Download-Paket Kinderprogramm.

Ein Bild kann in mehreren Gruppen vorhanden sein. Deshalb enthält dieses Feld einen Binärwert, der sich aus der Addition der verschiedenen Werte der Gruppen ergibt.

PI_MATERIAL

Feld	Beschreibung
MAT_KEYi	eindeutiger Schlüssel für das Material
MAT_SENDERi	eindeutige Zuordnung zu einem Sender
MAT_MANDANTi	eindeutige Zuordnung zu einem Mandanten
MAT_TYPi	Verweis auf einen Eintrag im Wertebereich für "Material"
MAT_PFADvc	gibt den Pfad an, unter dem die Materialdateien abgelegt worden sind
MAT_SENDEDATUMd	Ausstrahlungsdatum der Sendung, auf die sich das Material bezieht
MAT_STARTZEITd	Startzeit der Sendung auf die sich das Material bezieht
MAT_SENETITELvc	Sendetitel der Sendung auf die sich das Material bezieht
MAT_KWi	Kalenderwoche der Sendung auf die sich das Material bezieht
MAT_INFORMATIONvc	nähere Information zum Material
MAT_IMPORTDATUMd	Einfügedatum des Materials in diese Tabelle
MAT_GELOESCHT_KZb	Kennzeichen, ob das Material noch gültig ist
MAT_PAKET_TYPi	Verweis auf Einträge im Wertebereich für "Paket"
TIMESTAMP	

Wie schon vorher erwähnt, muß jeder Eintrag in der Materialtabelle eine Zuordnung haben, um in der Programmwoche dargestellt zu werden. Im folgenden wird der Ablauf einer Verbindung zu einem Element der Programmwoche erläutert. Die Zuordnung, welches Material zu welchem Sendepfad gehört, geschieht über die Material-Programmwochen-Tabelle. Auf der Materialtabelle ist ein Insert-Trigger angelegt. Sobald ein Eintrag in diese Tabelle erstellt wird, versucht die Datenbank automatisch über den Sendetag und die Startzeit den passenden Eintrag in der Programmwochentabelle zu finden. Der Trigger sucht in einem Zeitraum von +/- 15 Minuten der angegebenen Startzeit. Wenn genau ein passender Eintrag in der Programmwoche gefunden wird, wird in der Material-Programmwochen-Tabelle eine Zuordnung erstellt. Findet der Trigger kein oder mehr als ein Objekt in der Programmwochentabelle, das in den angegebenen Zeitraum paßt, wird kein Eintrag erstellt.

Das Material ohne eine Referenz muß vom Redakteur manuell zugewiesen werden. Dafür hat er zur Vereinfachung der Suche die Grundinformationen - Programmwoche, Sendetag, Startzeit und Sendetitel - für jedes Material. Beim Löschen überprüft ebenfalls ein Trigger, ob noch Einträge in den Tabellen PI_MATERIAL_PROGRAMMWOCHE und PI_MATERIAL_PERSON vorhanden sind. Nur wenn keine Beziehungen mehr vorhanden sind, kann ein Eintrag aus der Materialtabelle gelöscht werden.

PI_MATERIAL_PROGRAMMWOCHE

Feld	Beschreibung
MAPW_PW_KEYi	eindeutige Zuordnung mit einem Eintrag in der Programmwochentabelle
MAPW_MAT_KEYi	eindeutige Zuordnung mit einem Eintrag in der Materialtabelle

Beim Einfügen oder Ändern der Schlüssel eines Eintrages dieser Tabelle wird über einen Trigger geprüft, ob es die eingetragenden Schlüssel in den Tabellen PI_MATERIAL und PI_PROGRAMMWOCHE gibt. Schlägt die Überprüfung fehl, wird der Eintrag nicht in die Tabelle eingefügt.

Die Material/Person-Tabelle bildet die Zugehörigkeit von Personen zu Material (Bilder, Pressemappe, etc.) ab. Die Zuordnung geschieht über die Felder Sender, Mandant, Materialschlüssel und Personenschlüssel. Bisher findet diese Tabelle aber noch keine Verwendung innerhalb der Programmwoche. Ziel ist, in nächster Zukunft auch Bildmaterial zu einzelnen Personen über diese Datenbank zu verwalten. Der Benutzer des B2B-Angebotes kann sich dann nicht nur Detailinformationen zu einer Person ansehen, sondern auch zeitgleich das dazugehörige Bildmaterial.

PI_MATERIAL_PERSON

Feld	Beschreibung
MAPE_PERS_KEYi	Verweis auf eine Person zusammen mit der Sender- und Mandanten-ID
MAPE_SENDERi	eindeutige Zuordnung zu einem Sender
MAPE_MANDANTI	eindeutige Zuordnung zu einem Mandanten
MAPE_MAT_KEYi	Verweis auf einen Eintrag in der Tabelle Material

Trigger verhindern Einträge in der Tabelle, die auf Personen oder Materialien zeigen, die nicht vorhanden sind.

7.2.5 Version

Über die Versionstabelle wird die Historie der veröffentlichten Programmwochen verwaltet. Jede Änderung des Programmplanes nach seiner Veröffentlichung bewirkt, daß eine neue Version erstellt wird. Dadurch kann jederzeit nachvollzogen werden, wie die Programmwoche zu einem früheren Zeitpunkt ausgesehen hat.

Die Versionen beziehen sich auf die einzelnen Wochen und werden für jeden Sender über die Sender- und Mandanten-ID separat gepflegt. Die Versionsnummer beginnt bei 1 und wird bei jeder neuen Version für eine Programmwoche um 1 hoch gezählt. Mit Hilfe dieser Nummer wird auch innerhalb der Programmwoche der Gültigkeitszeitraum über die Felder "Gültig von"-Version und "Gültig bis"-Version abgegrenzt.

Über das Export- und Ablaufdatum wird der zeitliche Raum abgegrenzt, während dem die Version gültig ist. Eine Version, die noch keine Daten in den oben genannten Feldern enthält, wurde noch nicht veröffentlicht. Besitzt sie nur den Eintrag im Feld Exportdatum, so beinhaltet diese den aktuellen Programmplan. Ab dem Zeitpunkt, an dem eine Version ein Ablaufdatum hat, gibt es mindestens eine neuere Programmwoche.

PI_VERSION

Feld	Beschreibung
VER_Kwi	Programmwoche für diese Version
VER_SENDERi	eindeutige Zuordnung für einen Sender
VER_MANDANTI	eindeutige Zuordnung für einen Mandanten
VER_VERSIONi	Versionsnummer
VER_EXPORTDATUMd	Datum und Zeit der ersten Veröffentlichung unter dieser Version
VER_ABLAUFDATUMd	Datum und Zeit des Ablaufes dieser Version

7.3 Änderungen

7.3.1 Elementare Änderungen

Sämtliche Änderungen der Programmwoche zwischen der initialen Füllung der Datenbank bis zum Ablauf der Programmwoche werden in der Änderungstabelle gesammelt. Die Einträge dieser Tabelle werden zum größten Teil über die Stored Procedure "AenderungsUeberpruefung" automatisch generiert, können aber im Notfall (eine Änderung wurde nicht erkannt, oder Systemausfall von TRAFFIC oder PROFIS) manuell eingetragen werden.

Jede Elementaränderung der Programmwoche erhält einen eindeutigen Schlüssel. Über einen Verweis auf ein Programmwochenobjekt bezieht sich ein Eintrag der Änderungstabelle immer auf eine Sendung. Dabei wird immer der Schlüssel des neuen Objektes in der Programmwoche benutzt. Dadurch ist es möglich, einen Verweis auf eine redaktionelle Änderung aus dem aktuellen Programmplan herzustellen.

Die Bemerkung einer elementare Änderung wird beim Änderungsabgleich automatisch generiert. Sie enthält z.B. den alte Sendetitel oder den der ausgetauschte Sendung. Bei elementaren Änderungen, die manuell eingegeben worden sind, steht in diesem Feld der Text, den ein Redakteur eingegeben hat.

Mit Hilfe eines Wertebereichs wird die Art der Änderung im Änderungstyp aufgeschlüsselt. Es wird zwischen 13 verschiedenen Änderungen differenziert, wobei nicht alle automatisch erkannt werden können.

ID	Änderung	Beschreibung	Automatisch erkannt
0	Wie geplant	für die Ausgabe einer kompletten redaktionellen Änderung	-
1	Programmänderung	Film, Serie oder Sonderablauf wurden durch eine andere Sendung ersetzt	ja
2	Zeitverschiebung	Startzeit einer Sendung hat sich geändert	ja
3	Titeländerung	Sendetitel hat sich geändert	ja
4	Episodenänderung	Episode einer Serie wurde ausgetauscht	ja
5	Themenänderung	Thema einer Talkshow hat sich geändert	nein (wird als Episodenänderung erkannt)
6	Zusatz	eine Sendung wurde dem Programmplan hinzugefügt	ja
7	Neues Bildmaterial	neue Bilder für eine Sendung	nein
8	Entfällt	eine Sendung wurde aus dem Programmplan ersatzlos gestrichen	ja

9	Titelzusatz	der Sendetitel hat eine Erweiterung zum alten Titel erhalten	nein (wird als Titeländerung erkannt)
10	Untertitelzusatz	der Kommentar zu einem Sonderablauf hat sich geändert	nein
11	Korrektur	Fehler in den redaktionellen Texten zu einer Sendung wurden korrigiert	nein
12	Nachtrag	Zusatzinformationen zu einer Sendung, die bei der vorherigen Veröffentlichung noch nicht vorlagen	nein

Im Datumsfeld wird der Tag und die Uhrzeit der Erstellung für elementare Änderungen festgehalten.

PI_AENDERUNG

Feld	Beschreibung
AEND_KEYi	eindeutiger Schlüssel
AEND_PW_KEYi	Verweis auf einen Eintrag in der Programmwoche
AEND_BEMERKUNGvc	nähere Informationen zur Änderung
AEND_TYPi	Art der Änderung
AEND_DATUMd	Erstellungs- oder Generierungsdatum
TIMESTAMP	

Beim Einfügen einer elementaren Änderung wird über einen Trigger geprüft, ob der Verweis auf dem Eintrag in der Programmwoche gültig ist. Zeigt dieser auf kein Objekt in der Programmwochentabelle, wird der Eintrag der Änderungstabelle wieder zurückgenommen. Ein Delete-Trigger verhindert das Löschen einer elementaren Änderung, wenn noch eine Zuordnung zu einer redaktionellen Änderung in der Tabelle PI_RELEVANTE_AENDERUNG besteht.

7.3.2 Redaktionelle Änderungen

Redaktionelle Änderungen fassen 1 bis n elementare Änderungen zu einem Kontext zusammen. Der Austausch eines Films hat meist zur Folge, daß sich die nachfolgenden Sendungen verschieben. Der Änderungsabgleich fügt für jede einzelne Änderungen ein

Element in die Änderungstabelle ein. Der Redakteur kann diese zu einer redaktionellen Änderung zusammenfassen.

Für die redaktionellen Änderungen gibt es zwei Tabellen.

In der Änderungshinweis-Tabelle werden die Informationen zu einer redaktionellen Änderung gespeichert. Diese erhält einen eindeutigen Schlüssel.

Der Verweis zu einer Änderungsmeldung wird ebenfalls im Änderungshinweiselement gespeichert. Sobald die redaktionelle Änderungen einer Meldung zugeordnet wird, erhält dieses Feld den dazugehörigen Schlüssel.

In den Bemerkungsfeldern stehen die Hintergrundinformationen, warum diese elementaren Änderungen zu einer redaktionellen Änderung zusammengefaßt worden sind. Da diese Bemerkungen länger als 254 Zeichen sein können, stehen dem Redakteur zwei Felder zur Verfügung.

Im Datumsfeld wird der Tag und die Uhrzeit der Erstellung einer redaktionellen Änderung festgehalten.

PI_AENDERUNGSHINWEIS

Feld	Beschreibung
HIN_KEYi	eindeutiger Schlüssel für eine redaktionelle Änderung
HIN_MELD_KEYi	Verweis auf einen Eintrag in der Tabelle der Änderungsmeldung
HIN_BEMERKUNG1vc	Feld für Hintergrundinformationen dieser redaktionellen Änderung
HIN_BEMERKUNG2vc	Zusätzliches Feld für Hintergrundinformationen dieser redaktionellen Änderung
HIN_DATUMd	Erstellungsdatum
TIMESTAMP	

Beim Löschen eines Eintrages aus dieser Tabelle wird über einen Trigger überprüft, ob noch ein Verweis für dieses Objekt in der Tabelle PI_RELEVANTE_ÄNDERUNG existiert. Nur wenn dies nicht der Fall ist, kann der Änderungshinweis gelöscht werden.

Über die Tabelle PI_RELEVANTE_AENDERUNG wird die Zuordnung zwischen elementaren und redaktionellen Änderungen geschaffen. Grundbedingung dabei ist, daß eine redaktionelle Änderung mindestens eine Elementaränderung beinhaltet. Die Zuordnung wird über die eindeutigen Schlüssel der beiden Objekte hergestellt.

PI_RELEVANTE_AENDERUNG

Feld	Beschreibung
REAND_HIN_KEYi	Verweis auf einen Änderungshinweis
REAND_AEND_KEYi	Verweis auf eine elementare Änderung

Beim Einfügen und Ändern der Zuordnung in dieser Tabelle wird über Trigger geprüft, ob die jeweiligen Schlüssel in den Tabellen PI_AENDERUNG und PI_AENDERUNGSHINWEIS vorhanden sind. Es können nur Beziehungen zwischen elementaren und einer redaktionellen Änderung hergestellt werden, wenn es die jeweiligen Schlüssel gibt.

7.3.3 Änderungsmeldungen

Die Änderungsmeldung beinhaltet alle redaktionellen Änderungen zwischen zwei Veröffentlichungen der Programmwoche.

Jede Änderungsmeldung hat einen eindeutigen Schlüssel. Dieser wird auch in allen redaktionellen Änderungen eingetragen, die zu dieser Meldung gehören.

Über Sender und Mandant sind die Meldungen einem Fernsehsender zugeordnet. Jede Änderungsmeldung erhält eine fortlaufende Nummer und das aktuelle Jahr als Kennung für die spätere Ausgabe. Für eine neue Nummer wird immer der höchste Eintrag für einen Sender und Mandanten in einem Jahr genommen und um eins inkrementiert.

PI_AENDERUNGSMELDUNG

Feld	Beschreibung
MELD_KEYi	eindeutiger Schlüssel für eine Änderungsmeldung
MELD_SENDERi	eindeutige Zuordnung zu einem Sender
MELD_MANDANTi	Eindeutige Zuordnung zu einem Mandanten
MELD_NUMMERi	laufende Nummer
MELD_JAHRI	aktuelles Jahr
MELD_DATUMd	Erstellungsdatum

Beim Löschen einer Änderungsmeldung wird überprüft, ob ebenfalls eine redaktionelle Änderung auf diese verweist. Sollte es noch Beziehungen geben, wird das Löschen in der Änderungsmeldung-Tabelle rückgängig gemacht.

7.4 Personen

In der Personentabelle werden die Daten aller Personen gespeichert, die in Sendungen mitwirken, die in der Programmwochen-Datenbank enthalten sind. Die Informationen stammen dabei aus PROFIS und werden in die Programmwochendatenbank übernommen.

Jede Person hat einen Schlüssel, der zusammen mit der Sender- und Mandanten-ID eindeutig ist. So ist es auch möglich, Änderungen in den Personendaten an die Basissysteme zurückzugeben.

Die Personendaten bestehen aus Vornamen, Nachname, Geburtstag, eventuell Todestag, Nationalität, Biographie, bisherige Filme, Anekdoten, Telefon- und Telefaxnummer, E-Mail-Adresse und Homepage.

Die meisten Informationen aus dieser Tabelle werden zur Zeit noch nicht für die Erstellung der Programmwoche benötigt. Für die Zukunft ist geplant, daß es im B2B-Angebot auch Details zu jeder Person, die in der Programmwoche enthalten ist, dargestellt werden können. Deshalb sind diese Felder im DB-Schema berücksichtigt.

Ein weiteres Attribut, das zu einer Person gehört, ist das Star-Kennzeichen. In den Sendungen der Programmwoche, in denen diese Person mitwirkt, wird diese Person besonders dargestellt.

PI_PERSON

Feld	Beschreibung
PERS_KEYi	eindeutige Zuordnung für eine Person für diesen Sender und diesen Mandanten
PERS_SENDERi	eindeutige Zuordnung zu einem Sender
PERS_MANDANTI	Eindeutige Zuordnung zu einem Mandanten
PERS_VORNAMEvc	Vorname(n)
PERS_NACHNAMEvc	Nachname
PERS_GEBURTSTAGd	Datum des Geburtstages
PERS_TODESTAGd	Datum des Todestages
PERS_NATIONALITAETi	Nationalität der Person (wird über den Wertebereich abgebildet)
PERS_BIOGRAPHIETx	Lebenslauf der Person
PERS_FILMOGRAPHIETx	bisherige Filme der Person
PERS_BEMERKUNGtx	verheiratet mit ... , sonstige interessante Nebensächlichkeiten
PERS_TELEFONvc	Telefonnummer der Person

PERS_TELEFAXvc	Telefaxnummer der Person
PERS_EMAILvc	E-Mail-Adresse der Person
PERS_HOMEPAGEvc	Adresse der Homepage einer Person
PERS_STARb	Starattribut einer Person
TIMESTAMP	

Beim Ändern der Attribute einer Person wird der neue Stand dieser Tabelle über einen Update-Trigger mit dem Eintrag im Basissystem abgeglichen. Beim Löschen eines Eintrages aus dieser Tabelle dürfen keine Beziehungen dieser Person zu den Tabellen für Stab, Darsteller und Material bestehen. Diese Überprüfung übernimmt ein Delete-Trigger.

Über die Darsteller-Tabelle werden die Schauspieler den Sendungen zugeordnet. Dafür wird der Bestands- und der Personenschlüssel zusammen mit der Sender- und Mandanten-ID gespeichert.

Der Rollenname und Größe des Auftritts in der Sendung wird ebenfalls in dieser Tabelle abgelegt.

PI_DARSTELLER

Feld	Beschreibung
DAR_BESTANDS_SCHLUESSELi	eindeutige Zuordnung zu einem Bestandsobjekt für diesen Sender und diesen Mandanten
DAR_SENDERi	eindeutige Zuordnung zu einem Sender
DAR_MANDANTI	eindeutige Zuordnung zu einem Mandanten
DAR_PERS_KEYi	eindeutiger Schlüssel zusammen mit der Sender- und Mandanten-ID für die Personentabelle
DAR_ROLLEvc	Rollenname in der zugehörigen Sendung
DAR_TYPI	Unterscheidung, ob der Schauspieler in dieser Sendung ein Haupt-, Nebendarsteller, Guest-Star ist oder einen Cameo-Auftritt hat. (wird über den Wertebereich aufgeschlüsselt)

Die folgende Tabelle bildet die Zugehörigkeit von Personen, die zum Stab gehören, zu einem Bestandsobjekt ab. Die Zuordnung zu einem Bestandsobjekt geschieht über den Bestandsschlüssel, Sender- und Mandanten-ID. Mit Hilfe des Personenschlüssels und der beiden ID's wird auf ein Objekt der Personentabelle verwiesen.

Die Funktion einer Person innerhalb des Stabes wird über das Feld Typ bestimmt. Aufgeschlüsselt über den Wertebereich, wird hier zwischen Regie, Drehbuch, Musik, Produzent, Kamera, Schnitt, Autor der Literaturvorlage und sonstigen Funktion unterschieden.

PI_STAB

Feld	Bemerkung
STAB_BESTANDS_SCHLUESSELi	eindeutige Zuordnung zu einem Bestandsobjekt für diesen Sender und diesen Mandanten
STAB_SENDERi	eindeutige Zuordnung zu einem Sender
STAB_MANDANTi	eindeutige Zuordnung zu einem Mandanten
STAB_PERS_KEYi	Verweis auf eine Person zusammen mit der Sender- und Mandanten-ID
STAB_TYPc	Funktion der Person innerhalb des Events: Schauspieler, Regie, Drehbuch etc. (wird über den Wertebereich aufgeschlüsselt)

Auf den Tabellen PI_DARSTELLER und PI_STAB liegen jeweils Trigger für Insert und Delete. Beim Einfügen eines Eintrages wird zunächst überprüft, ob die angegebene Bestands- und Personenschlüssel existieren. Neu erstellte Zuordnungen werden danach an das Basissystem weitergegeben. Das Löschen einer Verbindung zwischen einem Bestandsobjekt und einer Person wird ebenso über einen Trigger an das Basissystem weitergegeben und dort gelöscht.

7.5 Sonstige

7.5.1 Wertebereich

Der Wertebereich verwaltet alle Informationen für die Felder, die mit „Typ“ deklariert worden sind. Die Bereiche können für verschiedene Fernsehstationen variieren. Deshalb wird mit Hilfe der Sender- und Mandanten-ID zwischen denen unterschieden.

Die Wertebereiche für die verschiedenen Tabellen werden über das Feld „Name“ gruppiert. Ein Eintrag für eine Gruppe wäre z.B. „Genre“.

Die ID's werden im Feld „Wert“ und die Bedeutung dieser ID im Feld „Text“ gespeichert.

Im Feld „SUCHTEXT“ stehen Kürzel für den Text. Dies wird benötigt, um die Genres und Produktionsländer aus dem von PROFIS übergebenem String ausschlüsseln zu können.

PI_WERTEBEREICH

Feld	Beschreibung
WB_SENDERi	eindeutige Zuordnung für einen Sender
WB_MANDANTi	eindeutige Zuordnung für einen Mandanten
WB_NAMEvc	Name der Wertebereichsgruppe
WB_WERTi	ID eines Eintrages
WB_TEXTvc	Bedeutung der ID
WB_SUCHTEXTvc	Suchkürzel

7.5.2 Schlüsselverwaltung

Die Objekt_Key-Tabelle verwaltet alle selbstgenerierten Schlüssel der Programmwochendatenbank. Im Feld BEREICHS_KEY steht ein Index für den Bereich, für den der Schlüssel verwaltet werden soll. In BEZEICHNUNG wird dieser Bereich benannt.

NAECHSTE_KEY verwaltet, welcher der nächste Schlüssel ist und mit OBJ_STEPi wird die Schrittweite zum nächsten Schlüssel angegeben.

Diese Tabelle wird nur über eine Stored Procedure angesprochen. Diese gibt den nächsten Schlüssel für einen Bereich zurück und sorgt dafür, daß in der Tabelle der nächste Schlüssel auch richtig gesetzt wird.

PI_OBJEKT_KEY

Feld	Beschreibung
OBJ_BEREICHS_KEYi	Nummer des Bereichs
OBJ_BEZEICHNUNGvc	Name des Bereichs
OBJ_NAECHSTE_KEYi	nächster Schlüssel für diesen Bereich
OBJ_STEPi	Schrittweite zum nächsten Schlüssel

7.5.3 Programmversionsverwaltung

Alle Programme, die auf dieser Programmwochen-Datenbank arbeiten sollen, müssen in dieser Tabelle eingetragen sein. Sobald sich der Client auf der Datenbank anmeldet, wird zuerst überprüft, ob der Programmname zusammen mit der Versionsnummer in der Tabelle eingetragen ist. Ansonsten darf der Client nicht auf der Datenbank arbeiten.

So wird gesichert, daß kein veralteter Client Schaden in den Tabellen verursachen kann.

Die Versionsnummer unterteilt sich in vier Abschnitte. Anhand dieser Nummer können die Änderungen zur Vorgängerversion erkannt werden. Wurde die vierte Stelle der Nummer hoch gezählt, so wurde nur der Client geändert. Hat sich die dritte Stelle geändert, wurde am Datenbankmodell eine Veränderung vorgenommen. Die zweite Stelle wird hoch gezählt, wenn sich Client und Datenbank geändert haben. Die erste Stelle ändert sich nur, wenn es sich um ein komplett neues Programm handelt.

PI_PROGRAMMWOCHENVERSION

Feld	Beschreibung
PWVER_PROGRAMMNAMEvc	Name des Clients
PWVER_NR1ti	erste Stelle der Versionsnummer
PWVER_NR2ti	zweite Stelle der Versionsnummer
PWVER_NR3ti	dritte Stelle der Versionsnummer
PWVER_NR4ti	vierte Stelle der Versionsnummer
PWVER_BENUTZERvc	Benutzergruppe des Clients
PWVER_KOMMENTARvc	Bemerkungen zum Client

7.6 Stored Procedures

Die Stored Procedures realisieren die Aspekte der Client-Server-Architektur auf der Datenbank. Damit wird versucht, die Aktionen weitestgehend serverseitig ausführen zu lassen und den Client zu entlasten.

Auf der Datenbank befinden sich eine Vielzahl von Stored Procedures, die den Client in seiner Arbeit unterstützen. In diesem Kapitel wird nur auf die drei wichtigsten eingegangen:

- GetAktuelleProgrammwoche
- AenderungsUeberpruefung
- VeroeffentlichePW

Ein komplette Übersicht aller vorhandenen Stored Procedures und deren Funktionen befindet sich im Anhang unter Kapitel 11.6.

7.6.1 GetAktuelleProgrammwoche

Über diese Funktion wird der eingeplante Programmablaufplan einer ausgewählten Woche für einen bestimmten Sender initial in die Programmwochen-Datenbank übernommen.

Aufgerufen wird diese Procedure mit der Sender- und Mandanten-ID des Fernsehsenders und der zu importierenden Programmwoche.

Im ersten Schritt wird überprüft, ob die ausgewählte Woche schon importiert worden ist. Wird kein Eintrag in der Versionstabelle gefunden, erstellt die Stored Procedure die erste Version mit der Kalenderwoche, dem Sender und den Mandanten. Das Export- und Ablaufdatum werden mit Null-Werten gefüllt.

PI_VERSION

Feld	Eintrag 1
VER_Kwi	17
VER_SENDERi	1
VER_MANDANTi	1
VER_VERSIONi	1
VER_EXPORTDATUMd	NULL
VER_ABLAUFDATUMd	NULL

Jetzt wird für die angegebene Programmwoche der Programmablaufplan aus TRAFFIC ausgelesen. Einschränkungen sind dabei:

- nur Sendungen des Typs Film, Episode und Sonderablauf werden ausgelesen; Grund: in TRAFFIC werden auch alle Werbespots eingeplant
- die Iststartzeit einer Sendung darf nicht hinter der Startzeit des Sendetages liegen; Grund: bei Live-Events kann nicht immer abgeschätzt werden, wie lange die Sendung dauern wird (z.B. Boxen). Deshalb wird immer ein Notprogramm angelegt. Dieses soll aber nicht in der Programmwoche erscheinen und wird somit ausgeschlossen.

Die ausgelesenen Daten werden dann über einen Cursor einzeln abgearbeitet und in die Datenbank geschrieben. Bevor ein Eintrag in der Programmwochentabelle erstellt werden kann, muß das entsprechende Bestandsobjekt vorhanden sein. Deshalb wird überprüft, ob die Sendung schon in der Bestandsobjekt-Tabelle vorhanden ist. Wenn die Sendung nicht gefunden wird, werden die entsprechenden Daten aus PROFIS ausgelesen und das Bestandsobjekt in der Programmwoche angelegt. Im Programmwocheneintrag wird ein entsprechender Verweis über den Bestandsschlüssel zu diesem Objekt eingetragen.

Alle Einträge in der Programmwoche (innerhalb der initialisierten KW) erhalten im Feld PW_GUELTIG_VON_VERSIONi den Eintrag der ersten Version (1) und in PW_STATUSi ein Kennzeichen, daß dieses Objekt noch zu bearbeiten ist.

Wenn alles abgearbeitet ist, wird zum Client der Return-Value 0 zurückgegeben, um zu zeigen, daß die Aktion erfolgreich war.

7.6.2 AenderungsUeberpruefung

Die Änderungsüberprüfung wird mit der Sender- und Mandanten-ID aufgerufen. Sie vergleicht den Datenbestand der Programmwochen-DB mit den aktuell eingeplanten Daten von PROFIS und TRAFFIC. Bei Änderungen wird die Programmwochendatenbank aktualisiert und die dazugehörigen elementaren Änderungen werden geschrieben.

Um den Änderungsabgleich durchzuführen, werden zunächst alle benötigten Felder beider Programmpläne in temporäre Tabellen geschrieben (#PW_BUFFER, #PROFTRAF_BUFFER).

#PW_BUFFER	#PROFTRAFF_BUFFER
PW_Key	
PW_Planungsobjekt_ID	Planungsobjekt_ID
PW_Sendetag	Sendetag
PW_Startzeit	Startzeit
PW_Typ	Typ
PW_Bestands_Schluessel	Bestands_Schluessel
PW_Bestands_SchluesselSerie	Bestands_SchluesselSerie
PW_Sendetitel	Sendetitel
PW_SendetitelSerie	SendetitelSerie

Als erstes werden mit Hilfe eines Cursors die Einträge der Tabelle #PROFTRAFF_BUFFER durchgegangen und nach dem entsprechenden Eintrag in der Programmwoche gesucht. Dazu dient im ersten Schritt die Planungsobjekt_ID. Diese bleibt für ein Objekt in TRAFFIC solange erhalten, bis ein Eintrag gelöscht wird. Da diese ID auch in der Programmwochentabelle gespeichert wird, können die passenden Einträge aus beiden Systemen gesucht und auf Unterschiede verglichen werden.

Dabei können folgenden Änderungen erkannt werden:

- Programmänderung
- Zeitverschiebung
- Titeländerung
- Episoden bzw. Themenänderung

Anhand eines Beispiels ist die Arbeitsweise der Stored Procedure gezeigt.

Programmänderungen

Einträge mit gleicher Planungsobjekt_ID werden als erstes auf eine Programmänderung hin überprüft. Dafür werden bei Filmen und Sonderabläufen die Bestandsschlüssel und bei Serien die BestandsschlüsselSerie miteinander verglichen. Unterschiedliche Bestandsschlüssel weisen darauf hin, daß eine Sendung ausgetauscht worden ist.

Veränderung der Datenbank beim Austausch eines Filmes:

#PW_BUFFER

PW_Key	PW_Planungsobjekt_ID	PW_Sendetag	PW_Startzeit	PW_Typ	PW_Bestands_Schlüssel	PW_Bestands_SchlüsselSerie	PW_Sendetitel	PW_SendetitelSerie
1085	5819232	22.04.00	22.00	3	3811		RAMBO II - DER AUFTRAG	
1087	5806025	22.04.00	22.59	5	41622		RTL aktuell Newsflash (Mo-Fr 13.00-18.45 Uhr + Wochenenda)	

#PROFTRAFF_BUFFER

Planungsobjekt_ID	Sendetag	Startzeit	Typ	Bestands_Schlüssel	Bestands_SchlüsselSerie	Sendetitel	SendetitelSerie
5819232	22.04.00	22.00	3	47476		SPECIALIST, THE	
5806025	22.04.00	22.59	5	41622		RTL aktuell Newsflash (Mo-Fr 13.00-18.45 Uhr + Wochenenda)	

Veränderung der Datenbank beim Austausch einer Serie:

#PW_BUFFER

PW_Key	PW_Planungsobjekt_ID	PW_Sendetag	PW_Startzeit	PW_Typ	PW_Bestands_Schlüssel	PW_Bestands_SchlüsselSerie	PW_Sendetitel	PW_SendetitelSerie
1090	5891590	22.04.00	04.54	4	43622	43401	SCHUMME GÄSTE	Woody Woodpecker
1091	5891634	22.04.00	05.00	4	43685	43401	REICH, ABER GEIZIG	Woody Woodpecker
1092	5891561	22.04.00	05.06	4	46338	6699	DER TRICK MIT DEM SCHNEE	Darwin, der letzte Dinosaurier

#PROFTRAFF_BUFFER

Planungsobjekt_ID	Sendetag	Startzeit	Typ	Bestands_Schlüssel	Bestands_SchlüsselSerie	Sendetitel	SendetitelSerie
5891590	22.04.00	04.54	4	10080	8810	Blinky, ein Hausber?	Noozles, Die
5891634	22.04.00	05.00	4	43685	43401	REICH, ABER GEIZIG	Woody Woodpecker
5891561	22.04.00	05.06	4	46338	6699	DER TRICK MIT DEM SCHNEE	Darwin, der letzte Dinosaurier

Zeitverschiebung

Danach wird überprüft, ob sich die Startzeiten der Programmelemente verändert haben. Meistens tritt diese Änderung als Folge einer Programmänderung auf und die Startzeit der nachfolgenden Sendungen verschieben sich.

Vergleich auf unterschiedliche Startzeiten:

#PW_BUFFER

PW_Key	PW_Planungsobjekt_ID	PW_Sendetag	PW_Startzeit	PW_Typ	PW_Bestands_Schlüssel	PW_Bestands_SchlüsselSerie	PW_Sendetitel	PW_SendetitelSerie
1054	5024029	22.04.00	06.05	4	16084	6810	Eukalyptus Festival	Die Noozles
1053	5024093	22.04.00	06.27	4	16089	6810	Afrika und zurück	Die Noozles
1055	5023305	22.04.00	06.51	4	64499	42116	Die Gemse Laisl	Jim Henson's Animal Show

#PROFTRAFF_BUFFER

Planungsobjekt_ID	Sendetag	Startzeit	Typ	Bestands_Schlüssel	Bestands_SchlüsselSerie	Sendetitel	SendetitelSerie
5024029	22.04.00	06.05	4	16080	6810	Die große Überraschung	Noozles, Die
5024093	22.04.00	06.27	4	16089	6810	Afrika und zurück	Noozles, Die
5023305	22.04.00	06.45	4	64499	42116	Die Gemse Laisl	Jim Henson's Animal Show

Titeländerung

Jetzt werden die Elemente der Programmablaufpläne nach Änderungen im Sendetitel untersucht. Dabei wird überprüft, ob bei gleichem Bestandsschlüssel ein anderer Sendetitel existiert. Bei Serien wird zusätzlich noch der Serientitel geprüft.

#PW_BUFFER

PW_Key	PW_Planungsobjekt_ID	PW_Sendetag	PW_Startzeit	PW_Typ	PW_Bestands_Schlüssel	PW_Bestands_SchlüsselSerie	PW_Sendetitel	PW_SendetitelSerie
1089	5014899	22.04.00	03:18	3	1279		SONDEREINSATZ IN DER SCHLANGENBUCHT	
1090	5001590	22.04.00	04:54	4	43622	43401	SCHLIMME GÄSTE	Woody Woodpecker
1091	5001634	22.04.00	05:00	4	43669	43401	REICH, ABER GELDIG	Woody Woodpecker
1092	5001561	22.04.00	05:06	4	46338	6699	DER TRICK MIT DEM SCHNEE	Darwin, der letzte Dinosaurier

#PROFTRAFF_BUFFER

Planungsobjekt_ID	Sendetag	Startzeit	Typ	Bestands_Schlüssel	Bestands_SchlüsselSerie	Sendetitel	SendetitelSerie
5014899	22.04.00	03:18	3	1279		SONDEREINSATZ IN DER SCHLANGENBUCHT	
5001590	22.04.00	04:54	4	18080	8810	Glinky, ein Hausier?	Noozles, Die
5001634	22.04.00	05:00	4	43665	43401	REICH, ABER GELDIG	Woody Woodpecker
5001561	22.04.00	05:06	4	46338	6699	DER TRICK MIT DEM SCHNEE	Darwin, der letzte Dinosaurier

Episoden- bzw. Themenänderung

Bei Serien wird zusätzlich überprüft, ob eine andere Episode eingeplant worden ist. Dabei wird verglichen, ob sich bei gleichem BestandsschlüsselSerie der Bestandsschlüssel des Bestandsobjekts geändert hat.

#PW_BUFFER

PW_Key	PW_Planungsobjekt_ID	PW_Sendetag	PW_Startzeit	PW_Typ	PW_Bestands_Schlüssel	PW_Bestands_SchlüsselSerie	PW_Sendetitel	PW_SendetitelSerie
1054	5024029	22.04.00	06:05	4	16004	8810	Eukalyptus Festival	Die Noozles
1053	5024093	22.04.00	06:21	4	16069	8810	Afrika und zurück	Die Noozles
1055	5023305	22.04.00	06:51	4	64499	42118	Die Gemma Laisl	Jim Henson's Animal Show

#PROFTRAFF_BUFFER

Planungsobjekt_ID	Sendetag	Startzeit	Typ	Bestands_Schlüssel	Bestands_SchlüsselSerie	Sendetitel	SendetitelSerie
5024029	22.04.00	06:05	4	16080	8810	Die große Überraschung	Noozles, Die
5024093	22.04.00	06:21	4	16069	8810	Afrika und zurück	Noozles, Die
5023305	22.04.00	06:51	4	64499	42118	Die Gemma Laisl	Jim Henson's Animal Show

Immer wenn ein Objekt über die Planungsobjekt_ID gefunden wurde, werden diese aus den jeweiligen temporären Tabellen #PW_BUFFER und #PROFTRAFF_BUFFER gelöscht.

Nach dem ersten Durchlauf kann ein Rest von Sendungen in beiden Tabellen übrigbleiben mit unterschiedlichen Planungsobjekten, für die noch keine Zuordnung gefunden worden ist. Es muß jetzt innerhalb dieses Deltas unterschieden werden, ob es sich um Sendungen handelt, die in TRAFFIC durch Löschen und Einfügen eingeplant wurden (dadurch erhält ein Sendeplatz in TRAFFIC eine neue Planungsobjekt-ID), zusätzlich in den Programmplan aufgenommen wurden oder einfach entfallen sind.

Zusätzliche Programmänderung

Dafür werden zuerst mal die Einträge beider Tabellen über den Sendetag und der Startzeit (+/- 30 Minuten) miteinander verglichen. Im Beispiel wurde der Film ‚Der Dummschwätzer‘ durch ‚Einsam, Zweisam, Dreisam‘ ersetzt. Für gefundene, passende Einträge wird zuerst wiederum eine elementare Änderung erstellt und danach aus den temporären Tabellen gelöscht.

Zusatz

Alle Einträge, die in der Tabelle #PROFTRAF_BUFFER übrig bleiben, sind ‚Zusätze‘. Im Beispiel wurde die Serie ‚Spezialisten unterwegs‘ zusätzlich ins Programm aufgenommen.

#PW_BUFFER

PW_Key	PW_Planungsobjekt_ID	PW_Sendetag	PW_Startzeit	PW_Typ	PW_Bestands_Schluessel	PW_Bestands_SchluesselSerie	PW_Sendetitel	PW_SendetitelSerie
1079	5814047	22.04.00	15:52	3	74867		DUMMSCHWÄTZER, DER	
1081	5009682	22.04.00	17:33	3	3810		RAMBO III	
1083	5009627	22.04.00	19:09	5	54481		Explosiv - Weekend	
1082	5008910	22.04.00	20:02	5	73041		RTL aktuell - Weekend	
1084	5569479	22.04.00	20:15	4	92071	38150	Traumhochzeit 9900 - Folge 6 - EA 22.04.00	Traumhochzeit
1087	5806025	22.04.00	22:59	5	41622		RTL aktuell Newstflash (Mo-Fr 13.00-18.45 Uhr + Wochenende)	

#PROFTRAF_BUFFER

Planungsobjekt_ID	Sendetag	Startzeit	Typ	Bestands_Schluessel	Bestands_Schluessel Serie	Sendetitel	SendetitelSerie
6150549	22.04.00	15:54	3	74867		EINSAM, ZWEISAM, DREISAM	
6150552	22.04.00	17:00	4	13848	6728	CASINO FATAL	Spezialisten unterwegs
5009682	22.04.00	17:33	3	3810		RAMBO III	
5009627	22.04.00	19:09	5	54481		Explosiv - Weekend	
5008910	22.04.00	20:02	5	73041		RTL aktuell - Weekend	
5806025	22.04.00	22:59	5	41622		RTL aktuell Newstflash (Mo-Fr 13.00-18.45 Uhr + Wochenende)	

Gelöschte Objekte

Alle Einträge, die in der Tabelle #PW_BUFFER übrig bleiben, sind ‚Gelöschte Sendungen‘. Im Beispiel wurde die Sendung ‚Traumhochzeit‘ ersatzlos gestrichen.

Jede Änderung in der Programmwoche wird als elementare Änderung in der Tabelle PI_AENDERUNG protokolliert. Diese verweisen immer auf ein Eintrag in der Programmwoche. Bei Änderungen im Bestandsobjekt (z.B. Sendetitel) wird für jeden Eintrag in der Programmwoche, das sich diese Sendung bezieht, eine elementare Änderungen generiert.

Die Stored Procedure übernimmt zeitgleich auch den aktuellen Programmplan aus TRAFFIC. Dabei unterscheidet sie, ob es sich um die erste Änderung in dieser Version handelt oder es schon vorher welche gegeben hat.

Erste Änderung in einer Version

Dieser Fall tritt ein, wenn der Eintrag im Feld PW_GUELTIG_VON_VERSIONi kleiner der höchsten Version innerhalb der selben Kalenderwoche ist.

Der alte Eintrag in der Tabelle PI_PROGRAMMWOCHE erhält im Feld PW_GUELTIG_BIS_VERSIONi die Versionsnummer der letzten Version und ist somit nicht mehr für die nächste Veröffentlichung relevant. Zeitgleich wird ein neuer Eintrag mit den Attributen PW_GUELTIG_VON_VERSIONi der aktuellen Version und dem Status „zu bearbeiten“ erstellt.

Relevante Felder PI_PROGRAMMWOCHE	Alter Eintrag	Neuer Eintrag
PW_KEYi	132	256
PW_PLANUNGSOBJEKT_IDi	3442	3442
PW_NUMMERi	17	17
PW_SENDETAGd	22.04.2000	22.04.2000
PW_STARTZEITd	22.04.2000 12:30:00	22.04.2000 13:00:00
PW_BESTANDS_SCHLUESSELi	3545	3545
PW_GUELTIG_VON_VERSIONi	1	2
PW_GUELTIG_BIS_VERSIONi	1	Null
PW_STATUSi	4 (veröffentlicht)	1 (zu bearbeiten)

Nachfolgende Änderung in einer Version vor Veröffentlichung

Dieser Fall tritt ein, wenn das Feld PW_GUELTIG_VON_VERSIONi gleich dem Wert der höchsten Version der Kalenderwoche ist. Dieses hat keine Auswirkungen auf die aktuelle Version. Es wird lediglich ein Update auf das bestehende Objekt durchgeführt, um alle neuen Einträge einzufügen und der Status wird wieder auf „zu bearbeiten“ zurückgesetzt.

Relevante Felder PI_PROGRAMMWOCHE	alter Eintrag	Änderungen zu altem Eintrag
PW_KEYi	256	256
PW_PLANUNGSOBJEKT_Idi	3442	3442
PW_NUMMERi	17	17
PW_SENDETAGd	22.04.2000	22.04.2000
PW_STARTZEITd	22.04.2000 13:00:00	22.04.2000 13:00:00
PW_BESTANDS_SCHLUESSELi	3545	4535
PW_GUELTIG_VON_VERSIONi	2	2
PW_GUELTIG_BIS_VERSIONi	Null	Null
PW_STATUSi	2 (bearbeiten)	1 (zu bearbeiten)

Über den Returnvalue zeigt diese Procedure an, ob sie erfolgreich durchgelaufen ist. Ist dies der Fall, gibt sie eine 0 zurück. Sollte eines der beiden Basissysteme keine Daten an die Stored Procedure liefern, gibt sie eine 1 zurück.

7.6.3 VeroeffentlichePW

Die Procedure wird mit der Sender- und Mandanten-ID des Fernsehsenders und einem String, in dem die Programmwochen, die veröffentlicht werden sollen, durch Kommas separiert sind, aufgerufen.

Dieser String wird ausgelesen und die Nummern der Programmwochen in eine temporäre Tabelle namens #BufferPWs geschrieben. Dort wird für jede Nummer auch der Status vergeben. Dieser unterscheidet, ob es sich um eine initiale oder eine zusätzliche Veröffentlichung handelt. Beim Einfügen in die Tabelle wird davon ausgegangen, daß es sich bei allen um zusätzliche Veröffentlichungen handelt und der Status wird auf 0 gesetzt.

Bevor eine Programmwoche publiziert werden darf, muß eine Reihe von Bedingungen, die im folgenden über die Fehlercodes erklärt werden, erfüllt sein. Diese werden vorher überprüft und nur wenn alle Programmwochen alle Bedingungen erfüllt haben, können sie publiziert werden.

Deshalb gibt es neben der temporären Tabelle für die zu prüfenden Programmwochen zusätzlich noch eine Tabelle für alle gefundenen Fehler. Sie heißt #FehlerRueckgabe und unterteilt sich in drei Felder: Programmwoche, Fehlercode und Keys. Dort werden alle gefunden Fehler protokolliert und der Benutzer des Clients kann dementsprechend handeln.

Im folgenden werden die einzelnen Fehlercodes durchgearbeitet:

Fehlercode 1:

Die angegebene Programmwoche ist nicht auf der Datenbank vorhanden und kann deshalb auch nicht veröffentlicht werden. In der Fehlertabelle wird nur die Programmwoche und der Fehlercode angegeben.

Fehlercode 2:

Für die angegebene Programmwoche sind noch nicht alle Elemente im Status kontrolliert. Deshalb kann sie noch nicht veröffentlicht werden. Die Fehlertabelle enthält Einträge mit der Programmwoche, dem Fehlercode und allen Programmwochenschlüsseln, die nicht den Status 'kontrolliert' haben.

Fehlercode 3:

Nicht alle zugehörigen elementaren Änderungen der angegebenen Programmwoche wurden in redaktionellen Änderungen verarbeitet. Wenn eine Programmwoche veröffentlicht werden

soll, müssen dem Benutzer auch alle Änderungen zwischen dem letzten und jetzigen Stand angezeigt werden. Die Fehlertabelle erhält Einträge mit der Programmwoche und dem Fehlercode und allen Änderungsschlüsseln, die noch nicht verarbeitet worden sind.

Fehlercode 4:

Redaktionelle Änderungen dürfen keine elementaren Änderungen enthalten, die zu Programmwochen gehören, die bei diesem Aufruf der Procedure noch nicht mit veröffentlicht werden. Dadurch soll verhindert werden, daß Änderungshinweise schon vor der Veröffentlichung einer Programmwoche von außen sichtbar sind. Die Fehlertabelle enthält Einträge mit der Programmwoche, dem Fehlercode und allen Schlüsseln der redaktionellen Änderungen, die Querverbindungen zu noch nicht veröffentlichten Programmwochen enthalten.

Bei Programmwochen, die zum ersten Mal veröffentlicht werden, entfällt die Überprüfung auf die Fehlercodes 2 bis 4. Änderungen weisen auf Unterschiede zwischen zwei veröffentlichten Programmplänen hin. Dies ist bei der ersten Veröffentlichung nicht möglich. Deshalb wird jede Programmwoche entsprechend überprüft. Wenn für eine Programmwoche mit den übergebenen Sender und Mandanten die Tabelle PI_VERSION nur einen Eintrag mit der erste Version enthält, wird in der temporären Tabelle das Statusfeld für die entsprechende Programmwoche auf 1 gesetzt. Dadurch werden alle Überprüfungen der Stored Procedure auf Fehler in den elementaren Änderungen übersprungen.

Wenn nach allen Überprüfungen die Fehlertabelle keine Einträge hat, macht die Procedure weiter mit der Veröffentlichung der Programmwoche.

Alle Einträge der zu veröffentlichenden Programmwochen erhalten im Feld PW_STATUSi den Status ‚veröffentlicht‘. In der Tabelle PI_VERSION wird das Feld VER_EXPORTDATUMd für jede übergebene KW auf das aktuelle Datum gesetzt. Sollte es eine frühere Version geben, wird das Feld VER_ABLAUFDATUMd mit dem aktuellen Datum und der aktuellen Zeit gefüllt. Somit ist gewährleistet, daß es mehrere Versionen für einen Tag geben kann.

PI_VERSION

Feld	Eintrag 1	Eintrag 2
VER_KWi	17	17
VER_SENDERi	1	1
VER_MANDANTi	1	1
VER_VERISONi	1	2
VER_EXPORTDATUMd	01.04.2000 08:00:35	07.04.2000 10:03:24
VER_ABLAUFDATUMd	07.04.2000 10:03:23	NULL

Die relevanten Einträge einer aktuellen Programmwoche stehen in der Tabelle PI_PROGRAMMWOCHEN mit den Attributen PW_GUELTIG_BIS_VERSIONi = NULL und PW_STATUSi = ‚Veröffentlicht‘.

Bei einer initialen Veröffentlichung werden alle dazugehörigen elementaren Änderungen aus der Tabelle PI_ÄNDERUNG gelöscht.

Für alle anderen Programmwochen werden die redaktionellen Änderungen genommen und zu einer Änderungsmeldung zusammengefaßt.

Bei erfolgreicher Veröffentlichung der Programmwoche wird in der Fehlertabelle ein Eintrag ohne Programmwoche, aber mit Fehlercode = 0 und dem Schlüssel der Änderungsmeldung eingefügt.

Ein Recordset mit allen Einträgen der Fehlertabelle wird am Ende der Stored Procedure an den Client zurückgegeben.

7.7 Views

Die Tabellen der Basissysteme werden von Procedures und Triggern nie direkt aufgerufen. Es existiert auf der Programmwochendatenbank für jede Tabelle aus TRAFFIC und PROFIS ein entsprechender View, der benutzt wird. Die Namen der Views setzen sich aus den Tabellennamen aus dem Basissystem und einem ‚v‘ als Präfix zusammen. Vorteil dieser Methode ist, daß bei Tabellenänderungen in den Basissystemen nicht alle Procedures und Trigger überarbeitet werden müssen, sondern es muß nur der View angepaßt werden.

Mit Hilfe dieser Views werden die Daten aus den Basissystemen übernommen und Änderungen der redaktionellen Daten an PROFIS zurückgegeben. Dabei setzen die Views auf unterschiedliche Arten von Tabellen in TRAFFIC und PROFIS auf. Die Programmwochendatenbank liegt auf dem gleichen Server wie die PROFIS-Datenbank. Deshalb können die Views direkt auf die Tabellen zugreifen. Die TRAFFIC-Datenbank hat

einen eigenen Server. Aus diesem Grund werden die benötigten Tabellen in die Programmwochendatenbank repliziert und dann über einen View angesprochen.

8 Prototyp

8.1 Programmbeschreibung

Um die Funktionalität der Datenbank den Anwendern in den Programmwochen-Meetings näher zu bringen und um ein Beispiel für die Entwickler der späteren Client-Applikation zu haben, wurde ihm Rahmen dieser Projektarbeit ein Prototyp über ASP realisiert. Dieser zeigt die Basisfunktionalitäten, ohne auf Ergonomie Rücksicht zu nehmen:

- welche Möglichkeiten die Datenbank bereitstellt und
- wie könnte der Ablauf der Bearbeitung später aussehen.

Zum Ausführen des Prototyps müssen folgende Bedingungen erfüllt sein:

- Webservice

Da der Client eine Browseranwendung ist, wird ein Rechner mit installiertem Webservice benötigt. Dieser muß ASP-Skripte verarbeiten können und für jeden Benutzer erreichbar sein. Die Dateien des Prototypen müssen in ein Verzeichnis abgelegt werden, auf das der Webservice zugreifen darf. Für die Entwicklung dieses Clients wurde der Personal Web Server 4.0 von Microsoft verwendet. Der Server wurde lokal auf einem Arbeitsplatzrechner installiert. Auf diesem wurden auch die Seiten entwickelt. Über das Intranet konnte von jedem anderen Rechner der aktuelle Programmwochen-Client aufgerufen werden.

- Datenquelle anlegen

Da die ASP-Skripte über ein ADO-Objekt auf die Datenbank zugreifen, wird für das jeweilige Datenbanksystem ein OLEDB-Treiber benötigt. Dieser muß auf dem Rechner, auf dem der Webservice läuft, installiert sein. In diesem Fall wurde der Sybase OLEDB-Treiber verwendet. Nun kann eine Datenquelle definiert werden, auf die das ADO-Objekt zugreifen kann. In dieser Datenquelle wird angegeben, welcher Treiber benutzt werden soll. Ferner wird die Adresse des Datenbankservers eingetragen. Sollte sich die Datenbank oder das Datenbanksystem ändern, kann durch Anpassung der Datenquelle der Prototyp ohne großen Aufwand umgestellt werden.

- Bilder

Um Bilder importieren zu können, ist es wichtig, daß die Benutzer des Programmwochen-Clients lesenden Zugriff auf die Verzeichnisse der Bildwoche haben. Die Überprüfungsroutine beim Importieren, ob die Bilder der Bildwoche vorhanden sind,

geschieht lokal über das interne Netzwerk. Ist kein Zugriff für den Benutzer erlaubt, kann das Bildmaterial nicht überprüft und somit auch nicht importiert werden.

Der Prototyp kann von jedem Rechner benutzt werden, der auf die Netzwerkumgebung von RTL zugreift und einen Browser installiert hat. Die Leistung des Rechners und das benutzte Betriebssystem spielen dabei keine Rolle.

8.2 Benutzerhandbuch

Ausgangspunkt des Prototyps ist der Startbildschirm. Von hier aus kann in die Untermenüs

- Programmwoche bearbeiten,
- Material bearbeiten oder
- Personen bearbeiten

verzweigt werden.



Abbildung 26 Startmenü

8.2.1 Programmwoche bearbeiten

Im Untermenü der Programmwoche können folgende Aktionen ausgeführt werden:

- Programmwoche importieren
- Programmwoche bearbeiten
- Änderungsabgleich
- Änderungen bearbeiten
- Relevante Änderungen anzeigen
- Änderungsmeldungen anzeigen
- Verschiedene Versionen anzeigen

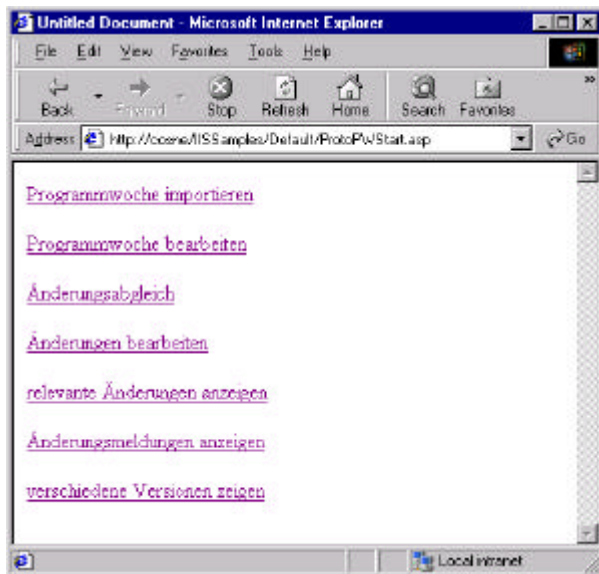


Abbildung 27 Startmenü Programmwoche bearbeiten

8.2.1.1 Programmwoche importieren

Zum Importieren der Programmwoche muß die gewünschte PW eingegeben werden. Der Import wird dann gestartet. Sollte der Import fehlerfrei verlaufen, ist die gewünschte Programmwoche in die Programmwochendatenbank übernommen worden.

8.2.1.2 Programmwoche bearbeiten

Zum Bearbeiten wird zuerst eine Programmwoche und danach ein Sendetag ausgewählt. Der Benutzer sieht den Programmablaufplan des ausgewählten Sendetages mit allen Programmelementen und deren Status.

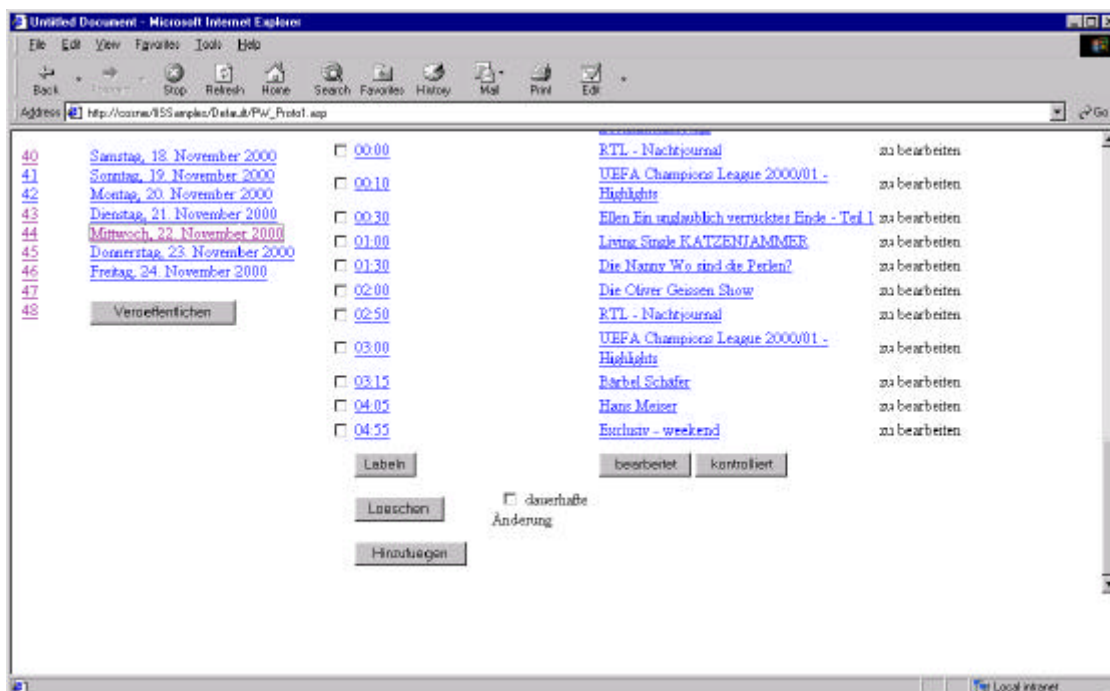


Abbildung 28 Programmablaufplan

8.2.1.2.1 Label

Im Programmablaufplan können verschiedene Sendungen über Checkboxes ausgewählt werden, für die Label zugeordnet werden soll. Über den Button 'Labeln' wird in diesen Teilbereich verzweigt.

Startbildschirm:

Im Startbildschirm werden alle Label angezeigt, die zu einer Sendung gehören. Der Benutzer erhält so einen Überblick über die vorhandenen Labels für einen Sendepplatz. Alle weiteren Aktionen, die im nachfolgenden besprochen werden, können von diesem Punkt aus aktiviert werden.

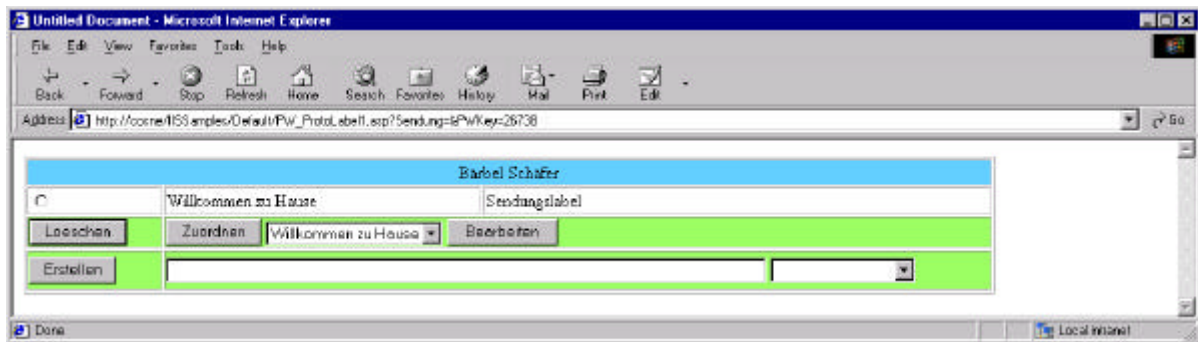


Abbildung 29 Label Übersicht

Erstellen eines neuen Labels:

Wenn für die ausgewählten Sendungen noch kein passendes Label vorhanden ist, kann dieses hinzugefügt werden. Dazu muß der gewünschte Text eingegeben und der Typ des Labels ausgewählt werden. Mit Hilfe des Buttons 'Hinzufügen' werden die eingetragenen Werte auf der Datenbank gespeichert und das neue Label steht ab sofort zur weiteren Verarbeitung zur Verfügung.

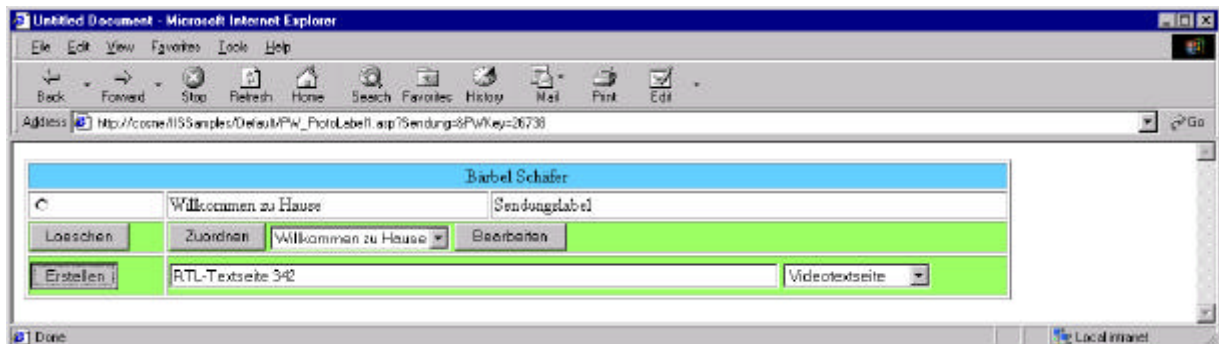


Abbildung 30 Erstellen eines Labeltextes

Zuordnen eines Labels:

Über die Listbox kann ein Label ausgewählt werden. Mit dem Button 'Zuordnen' werden für die ausgewählten Sendungen die entsprechenden Einträge in die Tabelle PI_Label geschrieben. Beim Versuch, das gleiche Label ein zweites Mal einer Sendung zuzuordnen, bricht das Programm mit einer Fehlermeldung ab.

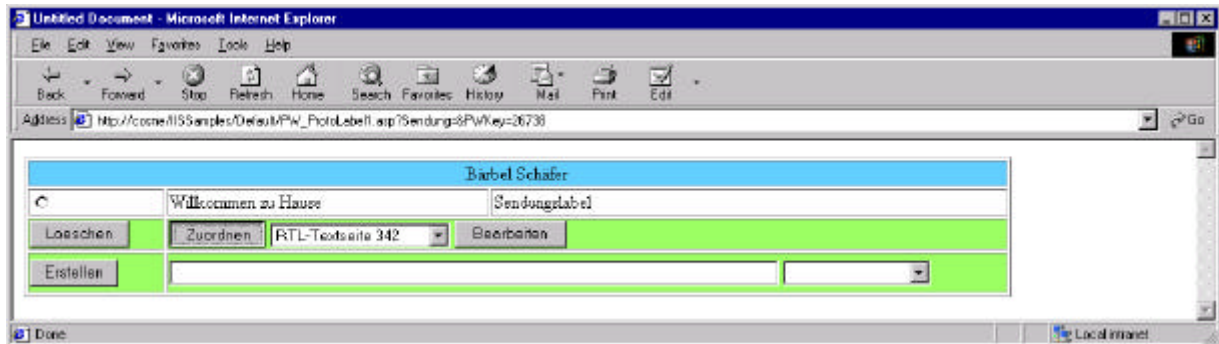


Abbildung 31 Zuordnen eines Labels zu einer Sendung

Verbindung löschen:

Durch Auswahl eines zugeordneten Labels über den Radiobutton und betätigen des "Löschen"-Buttons kann eine Zuordnung gelöscht werden.

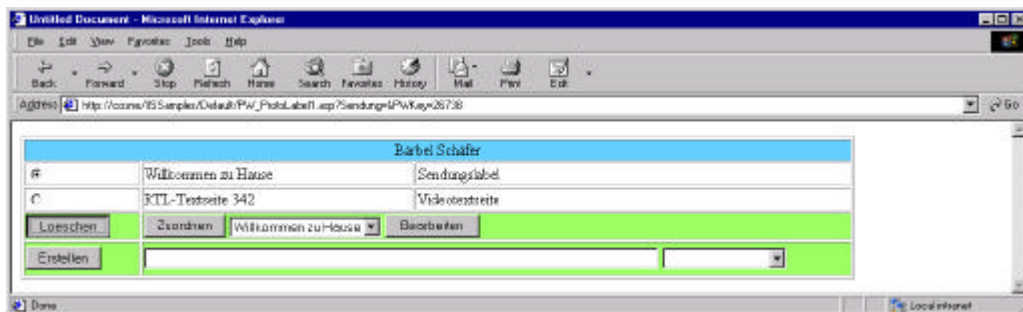
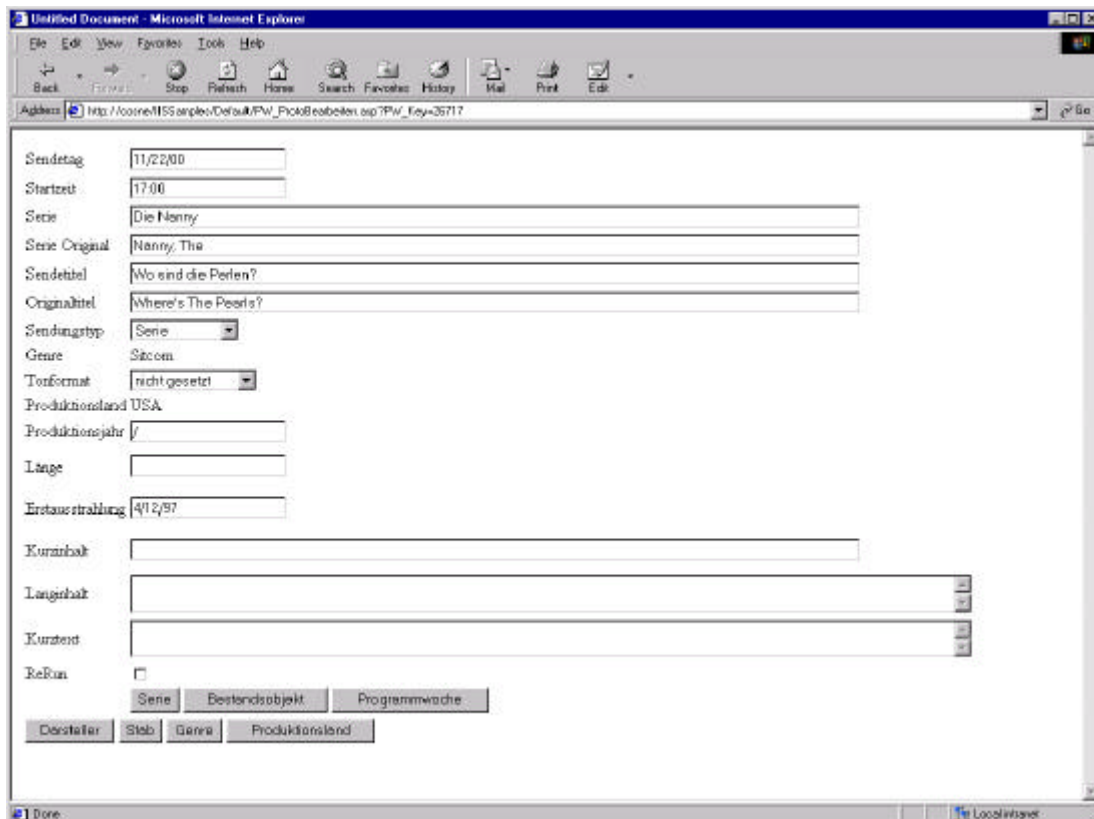


Abbildung 32 Löschen eines Labels

8.2.1.2.2 Bearbeitung des Programmablaufplans

Bearbeitung eines Programmwochenelements:

Jede Sendung innerhalb der Programmwoche kann separat ausgewählt und bearbeitet werden. Die Auswahl erfolgt durch Anklicken eines Elementes im Programmablaufplan. Der Benutzer springt in die Bearbeitungsmaske und kann hier die Attribute einer Sendung bearbeiten. Es können zum Beispiel redaktionelle Texte eingegeben, Sendetitel von Sendungen bzw. Serien verändert, das Genre bestimmt werden. Lediglich der Sendetag kann nicht bearbeitet werden.



The screenshot shows a web browser window with a form for editing a program element. The form contains the following fields and buttons:

- Sendetag: 11/22/00
- Startzeit: 17:00
- Serie: Die Nanny
- Serie Original: Nanny, The
- Sendetitel: Wo sind die Perlen?
- Originaltitel: Where's The Pearls?
- Sendungstyp: Serie
- Genre: Sitcom
- Tonformat: nicht gesetzt
- Produktionsland: USA
- Produktionsjahr: /
- Länge: /
- Erstausstrahlung: 4/12/97
- Kurzinhalt: /
- Langinhalt: /
- Kurztext: /
- ReRun:

Buttons at the bottom of the form include: Serie, Bestandsobjekt, Programmwoche, Darsteller, Stab, Genre, and Produktionsland.

Abbildung 33 Programmelement bearbeiten

Änderungen der Felder werden nach ihrer Zugehörigkeit zu den verschiedenen Datenbanktabellen gespeichert. Es wird unterschieden zwischen Serie, Bestandsobjekt und Programmwoche.

Folgende Bearbeitungsfelder werden mit den einzelnen Buttons abgespeichert:

- Serie: Serie, Serie Original
- Bestandsobjekt: Sendetitel, Originaltitel, Sendungstyp, Tonformat, Produktionsjahr, Länge und Erstausstrahlung
- Programmwoche: Startzeit, Kurzinhalt, Langinhalt, Kurztext und ReRun.

Diese Aufsplittung beim Speichern ist notwendig, da mehrere User gleichzeitig an einer Programmwoche arbeiten können. Sollten zwei Personen gleichzeitig an einem Eintrag der

Programmwoche arbeiten, muß verhindert werden, daß sie sich ihre Änderungen gegenseitig überschreiben können. Der Prototyp überprüft deshalb, ob sich seit dem Auslesen der Daten bis zum Zurückschreiben der Timestamp geändert hat. Damit ist sichergestellt, daß inzwischen keine Änderungen gemacht worden sind. Der Grund für die Untergliederung beim Speichern in drei Teilbereich soll an einem Beispiel verdeutlicht werden:

Zwei Redakteure arbeiten an zwei verschiedenen Episoden der gleichen Serie. Redakteur 1 schreibt einen redaktionellen Text für eine Episode, Redakteur 2 korrigiert gleichzeitig einen Fehler im Sendetitel der Serie. Wenn nun die kompletten Daten zu einem Element der Programmwoche gespeichert würden, könnte ein Redakteur seine Änderungen nicht speichern, obwohl beide Änderungen datenbankseitig und redaktionell nichts miteinander zu tun haben. Da es programmiertechnisch auch nicht möglich ist, abzufragen, welche Felder im Browserfenster bearbeitet worden sind, um dann dementsprechend zu reagieren, müssen Änderungen separat in der jeweiligen Tabelle gespeichert werden.

Genre und Produktionsland werden für eine Sendung gesondert in einer Auswahlliste zugeordnet. Nach Auswahl eines dieser beiden Buttons ("Genre" oder "Produktionsland") erhält der Benutzer ein Auswahlmenü mit alle Einträge, aus denen er für eine Sendung die entsprechenden auswählen kann.

Darsteller und Stäbe werden ebenfalls gesondert einer Sendung zugeordnet. Nach Auswahl eines dieser Buttons kommt der Benutzer in das folgende Untermenü.

Darsteller:

Beim Zuordnen einer Person als Darsteller muß zunächst der Rollename eingetragen und die Art des Auftritts in dieser Sendung (Hauptdarsteller, Nebendarsteller etc.) gewählt werden.

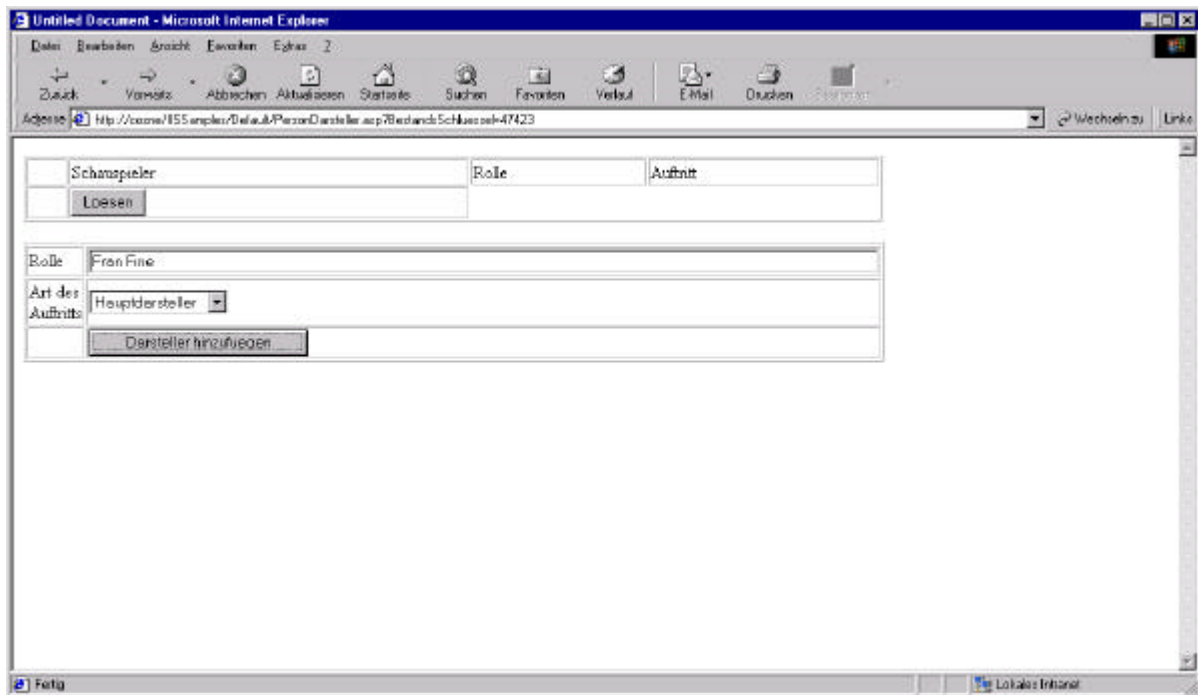


Abbildung 34 Neuen Darsteller zuordnen

Danach wählt der Anwender über eine Suchfunktion einen Schauspieler aus und stellt so die Verbindung zu einer Sendung her. Der Benutzer kann in diesen Dialog auch Personen hinzufügen oder die Personendaten bearbeiten. Darauf wird später in diesem Kapitel noch genauer eingegangen.

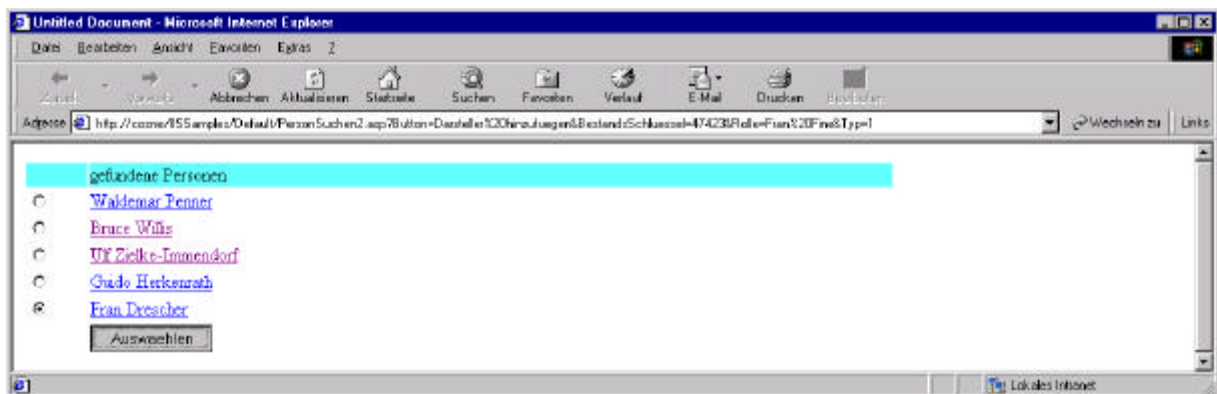


Abbildung 35 Auswahl einer Person

Der Benutzer kann sich alle Mitwirkenden einer Sendung anzeigen lassen. Durch Markierung des Radiobuttons und Anklicken des „Lösen“-Buttons kann eine Zuordnung wieder entfernt werden.

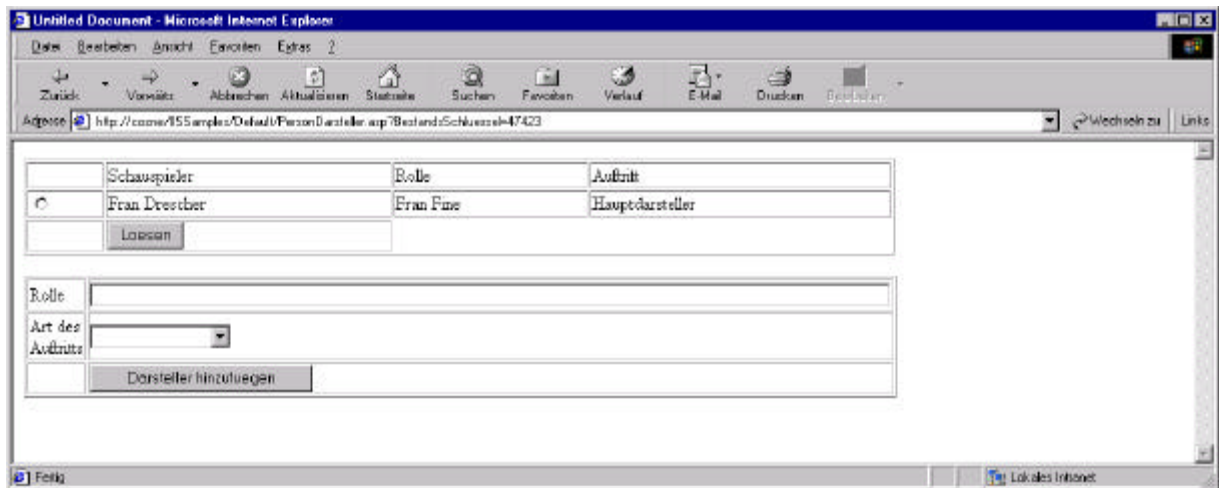


Abbildung 36 Darstellerliste

Stab:

Die Zuordnung des Stabes entspricht die der Darsteller. Es muß lediglich, anstatt der Rolle und der Art des Auftritts, die Funktion ausgewählt werden.

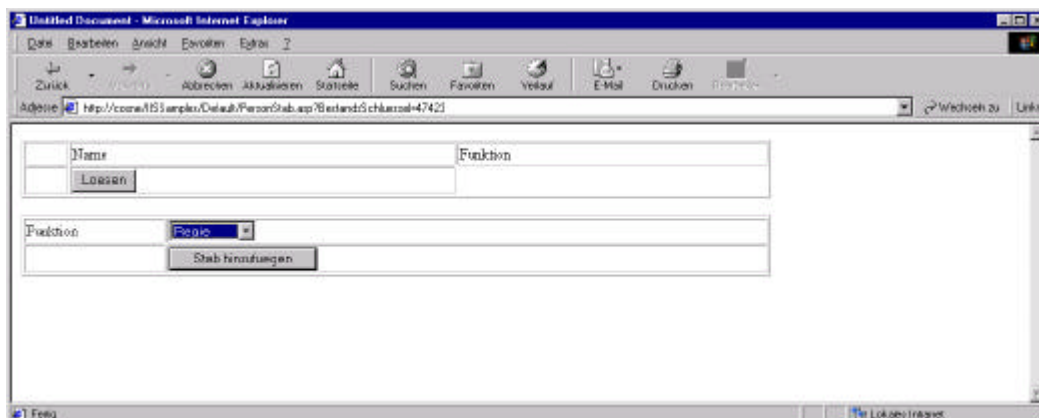


Abbildung 37 Stab zuordnen

Der spätere Client wird zusätzlich die Möglichkeit bieten, komplette Darsteller- und Stab-Listen zu kopieren

8.2.1.2.3 Verändern des Programmablaufplanes

Wenn eines der beiden zuliefernden Systeme (PROFIS, TRAFFIC) ausfällt, gibt es keine Möglichkeit, die Programmwoche - wie vorgesehen - zu generieren. Für diesen Fall gibt es eine Option, die es ermöglicht, Einträge in der Programmwoche manuell zu erstellen. Auch wenn Sendungen nicht in der Programmwoche erscheinen oder zusätzlich eingefügt werden sollen, ohne daß diese in TRAFFIC eingeplant sind, wird die manuelle Bearbeitung benutzt.

Hinzufügen eines manuellen Eintrages in der Programmwoche:

In der Ansicht des Programmablaufplans gibt es die Funktion ‚Hinzufügen‘. Über diesen Button können Objekte im Programmplan neu angelegt werden.

Der Benutzer erhält eine neue Bildschirmmaske, in der ein Programmwochenobjekt angelegt werden kann. Dafür muß die Startzeit angegeben und der Sendungstyp (Film, Serie, Sonderablauf) ausgewählt werden.

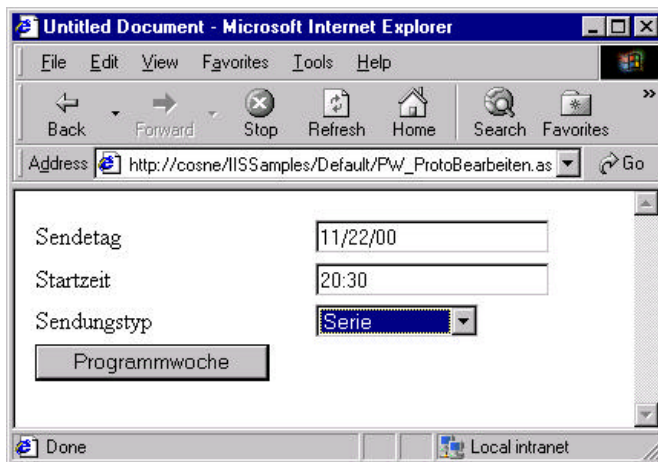


Abbildung 38 Hinzufügen eines Programmwocheneintrages

Das neu erstellte Objekt ist dann im Programmablaufplan verfügbar und kann zur weiteren Bearbeitung ausgewählt werden.

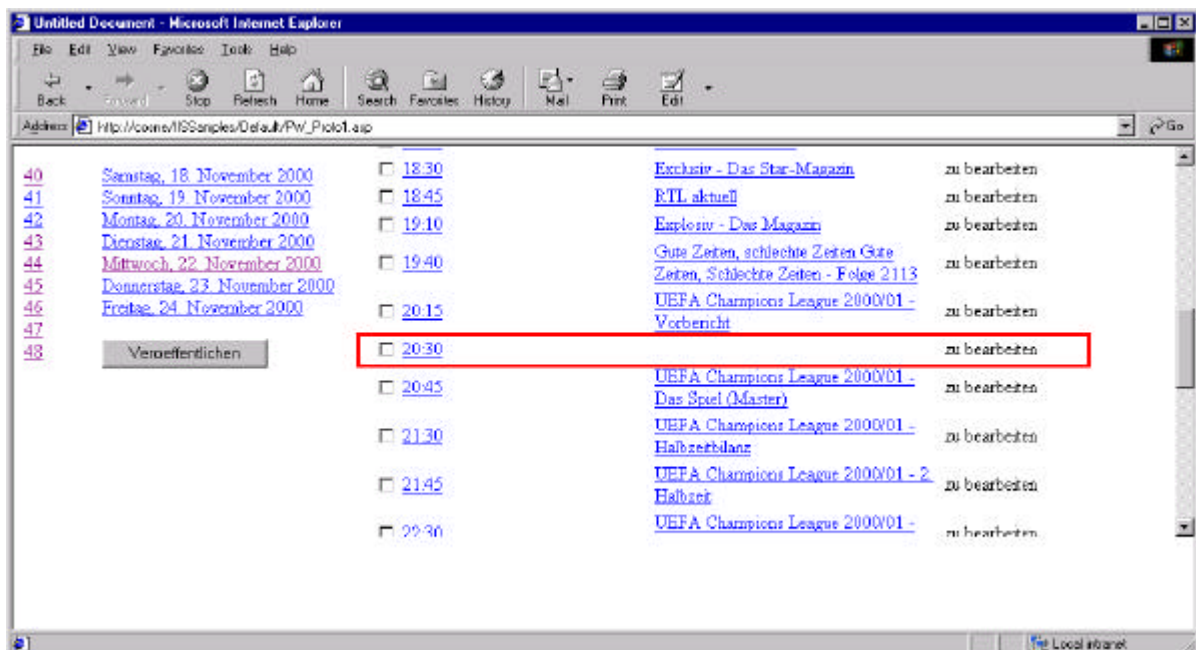


Abbildung 39 Manuell eingefügtes Programmwochenelement

Das Programmwochenelement kann jetzt genauso bearbeitet werden, wie zuvor beschrieben.

Löschen eines Eintrages:

In der Ansicht des Programmablaufplans eines Sendetages findet sich die Funktion ‚Löschen‘. Durch Auswahl der Checkbox neben einem Eintrag des Programmplans und betätigen des Buttons 'Loeschen' wird eine Sendung gelöscht.

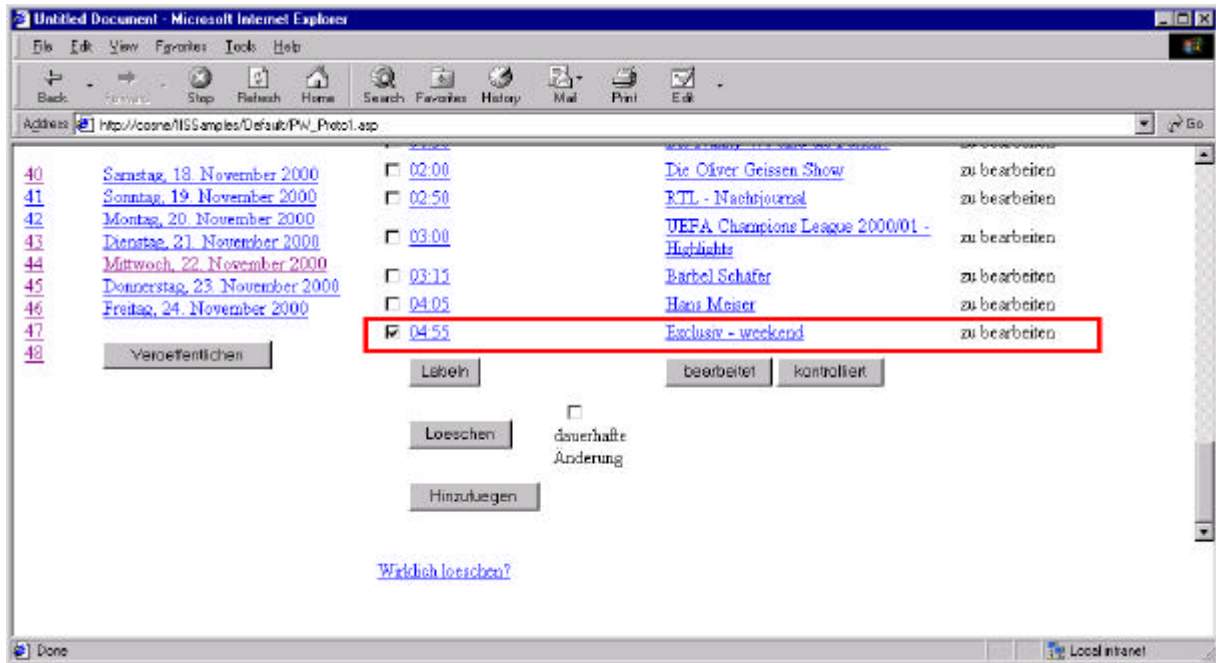


Abbildung 40 Löschen eines Programmwocheneintrages

Zur Sicherheit wird der Benutzer vor dem endgültigen Löschvorgang nochmals gefragt, ob er wirklich diese Aktion durchführen will. Nach Beenden dieses Vorganges, ist die Sendung für den Benutzer im Programmplan nicht mehr zu sehen. In der Datenbank wird lediglich in der Tabelle PI_PROGRAMMWOCHE das Gelöscht-Kennzeichen für dieses Objekt gesetzt.

Dauerhafte Änderung:

Bei den Aktionen Hinzufügen oder Löschen kann über eine Checkbox zwischen dauerhaften und nicht dauerhaften Änderung gewählt werden. Nach Abschluß einer dieser beiden Aktionen wird - je nach Auswahl - das Dauerhaft-Kennzeichen des Objekts in der Tabelle PI_PROGRAMMWOCHE auf 0 oder 1 gesetzt.

- Nicht dauerhafte Änderungen (flüchtige):

Die flüchtigen Änderungen stehen nur bis zum nächsten Abgleich mit PROFIS und TRAFFIC zur Verfügung. Sie sind dazu gedacht, bei einem Ausfall der zuliefernden Systeme eine aktuelle Programmwoche für das Internet generieren zu können. Wenn ein

neuer Abgleich gestartet wird, fallen alle manuellen Einträge raus, ohne daß ein Eintrag in die Tabelle PI_AENDERUNG erfolgt.

Um eine solche Aktion durchführen zu können, muß die Checkbox ‚Änderung dauerhaft‘ ungefüllt bleiben. Das Dauerhaft-Kennzeichen auf der Datenbank wird auf 1 gesetzt.

- Dauerhafte Änderungen (permanent):

Die permanenten Änderungen werden bei einem Abgleich nicht beachtet und bleiben bis zum Ablauf der Programmwoche bestehen. Sie werden benötigt, um Sendungen, die nicht in der endgültigen Programmwoche erscheinen sollen, aus dem Programmablaufplan herauszunehmen.

Um eine solche Aktion durchzuführen muß die Checkbox ‚Änderung dauerhaft‘ angekreuzt werden. Das Dauerhaft-Kennzeichen auf der Datenbank wird auf 0 gesetzt.

Status setzen (bearbeitet/kontrolliert):

Alle über die Checkbox ausgewählten Elemente der Programmwoche erhalten den Status des ausgewählten Buttons. Um eine Programmwoche zu veröffentlichen, müssen alle Einträge bearbeitet und kontrolliert sein.

Veröffentlichen:

Mit Hilfe dieser Aktion wird der Status der aktuell ausgewählten Programmwoche auf den Status 'veröffentlicht' gesetzt. Im Gegensatz zum produktiven Client wird in diesem Prototyp keine Rücksicht auf die Änderungen genommen, sondern nur überprüft, ob alle Elemente der Programmwoche mindestens den Status 'kontrolliert' erreicht haben (siehe Kapitel 7.6.3 VeroeffentlichePW). Es wird in der Versionstabelle für die angewählte Programmwoche das Exportdatum gesetzt und alle aktuellen Elemente im Programmablaufplan erhalten den Status 'veröffentlicht'.

8.2.1.3 Änderungsabgleich

Nach Auswahl dieses Unterpunktes kann der Benutzer den Änderungsabgleich über einen Button starten.

Sollte der Abgleich fehlerfrei verlaufen, ist der aktuelle Programmplan mit dem Stand aus PROFIS und TRAFFIC übernommen worden und jede Änderung in der Änderungstabelle protokolliert.

8.2.1.4 Änderungen bearbeiten

Auf der Startseite dieses Unterpunktes werden dem Benutzer alle Änderungen angezeigt, die noch nicht in einer redaktionellen Änderungen verarbeitet worden sind. Zugleich sind alle Aktionen aufgelistet, die mit einer Änderung durchgeführt werden können.

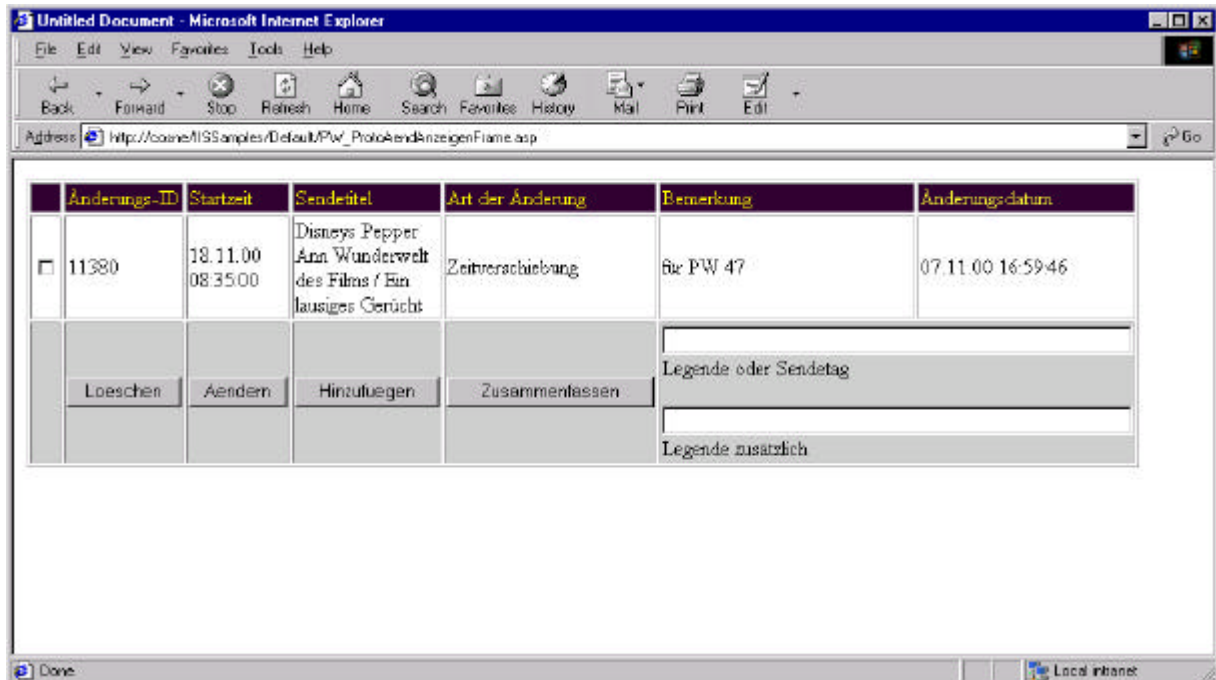


Abbildung 41 Übersicht aller elementaren Änderungen

Löschen:

Nicht alle elementaren Änderungen sollen später auch in der Programmwoche publiziert werden. Deshalb gibt es die Möglichkeit, einzelne Einträge über die Checkboxes auszuwählen und über den 'Löschen'-Button aus der Datenbank zu entfernen. Bevor diese Aktion ausgeführt wird, wird der Benutzer zu Sicherheit nochmal gefragt, ob diese Aktion wirklich durchgeführt werden soll.

Ändern:

Nicht alle Änderungen werden vom Änderungsabgleich richtig erkannt. So wird die Themenänderung einer Talkshow immer als Episodenänderung definiert. Deshalb kann jede elementare Änderung manuell nachbearbeitet werden. Dazu wird ein zu änderndes Element über die Checkbox ausgewählt und über den 'Ändern'-Button aktiviert.

Der Benutzer erhält eine Ansicht aller Eigenschaften dieser Änderung und kann die Art der Änderung sowie den Kommentar bearbeiten und speichern.

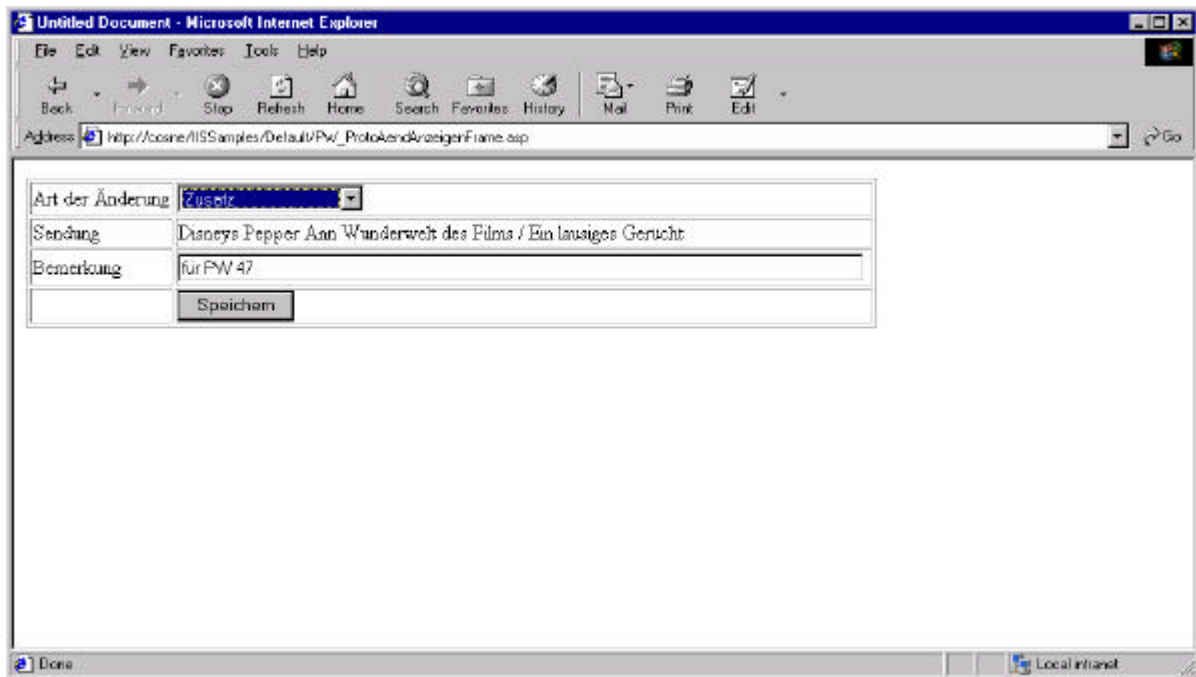


Abbildung 42 Attribute einer Änderung bearbeiten

Hinzufügen:

Manche Änderungen werden über den Änderungsabgleich nicht erfasst, müssen aber trotzdem in der Programmwoche ausgegeben werden. Ein Beispiel dafür sind Änderungen im Bildmaterial. Elementare Änderungen können deshalb auch manuell erstellt werden.

Dazu muß zuerst eine Sendung, zu der es eine Änderung gibt, im Programmablaufplan ausgewählt werden. Der Benutzer gibt den Sendetag ein und drückt den Button 'Hinzufügen'.

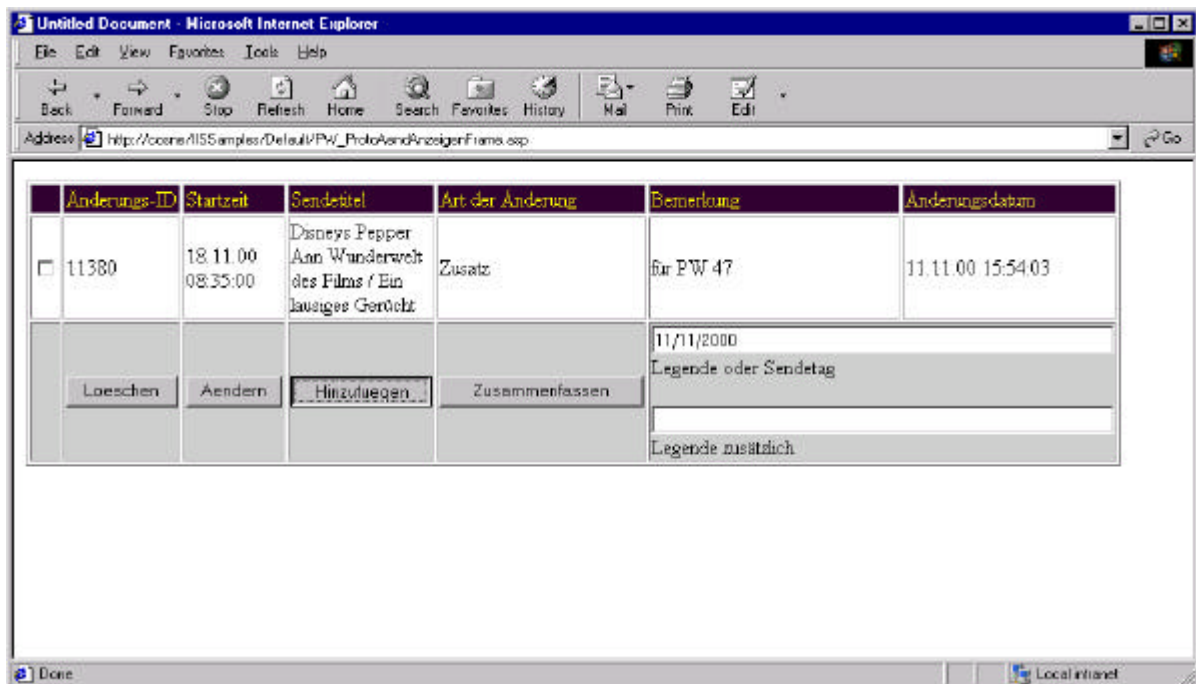


Abbildung 43 Änderung manuell hinzufügen

Die entsprechende Sendung kann jetzt aus dem Programmablaufplan ausgewählt werden.

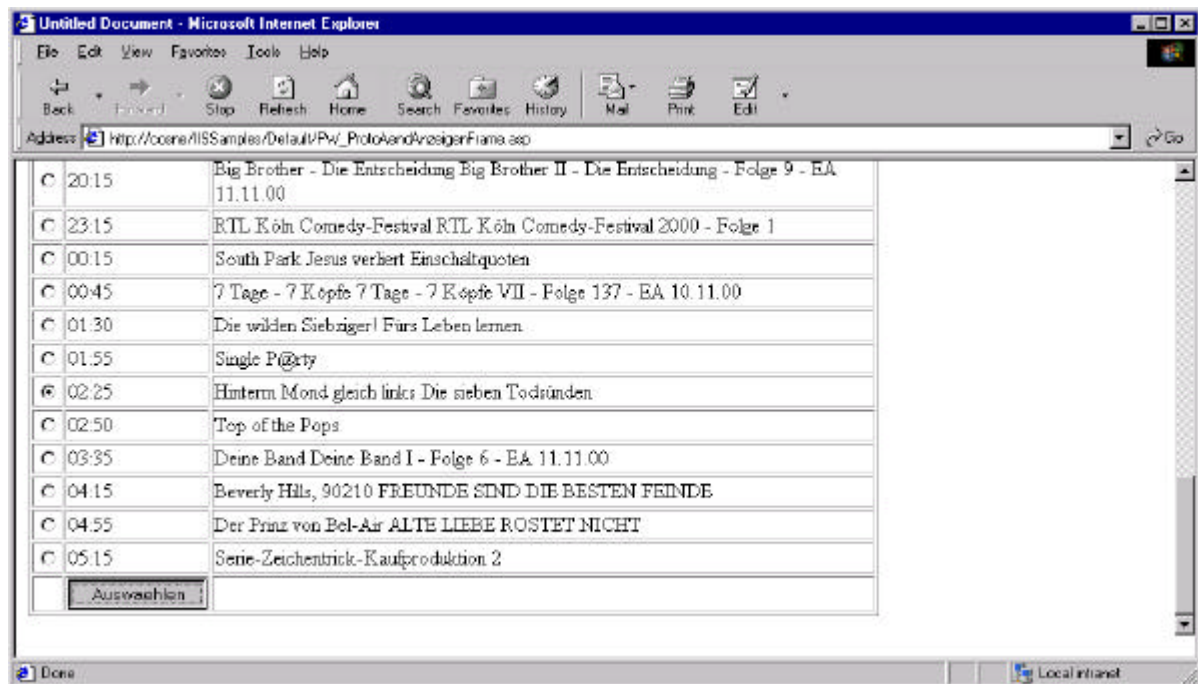


Abbildung 44 Auswahl einer Sendung

Es erscheint, wie beim Ändern einer elementaren Änderung, der Bearbeitungsdialog, in dem die Art der Änderung sowie der Kommentar bearbeitet und gespeichert werden muß.

Zusammenfassen:

1 bis n Änderungen können zu einer redaktionellen Änderung zusammengefaßt werden. Dafür werden alle elementaren Änderungen, die zusammen gehören, über Checkboxen ausgewählt, ferner wird eine Bemerkung dazu eingetragen. Diese Bemerkung erscheint bei der redaktionellen Änderung als Legende zu diesen Einträgen.

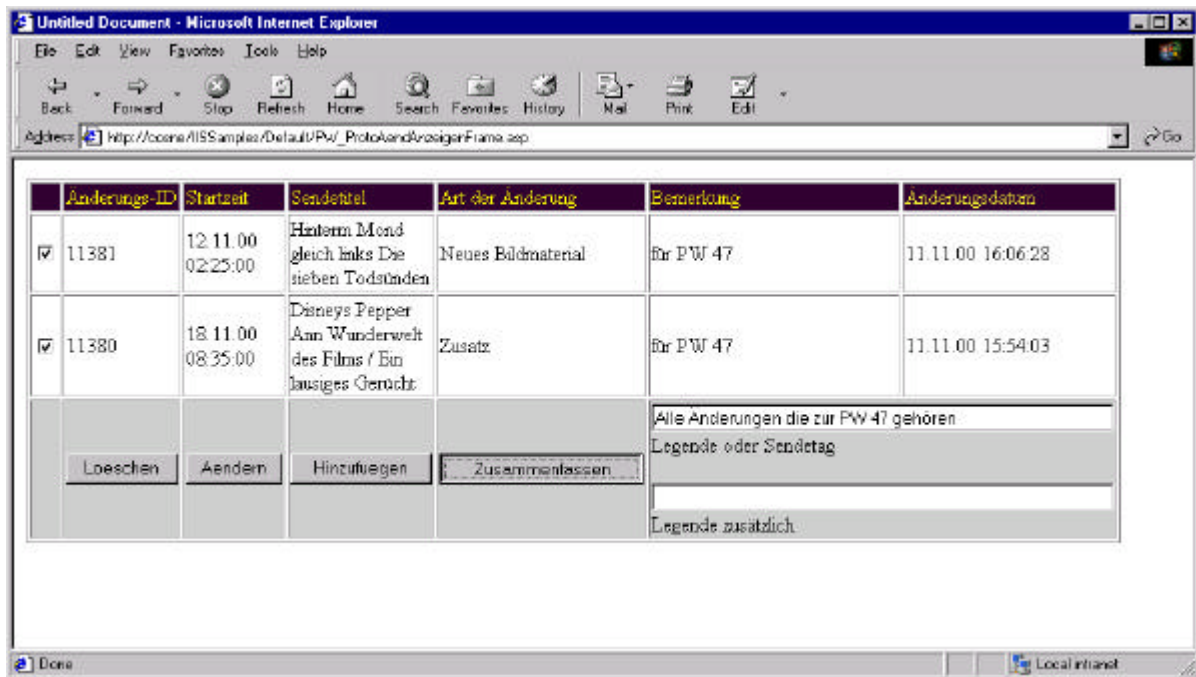


Abbildung 45 Redaktionelle Änderung erstellen

8.2.1.5 Relevante Änderungen anzeigen

Diese Ansicht zeigt alle redaktionellen Änderungen, sortiert nach der zugehörigen Programmwoche und Sendetag. Hier kann die Klammerung der redaktionellen Änderung um die einzelnen elementaren Änderungen überprüft werden.

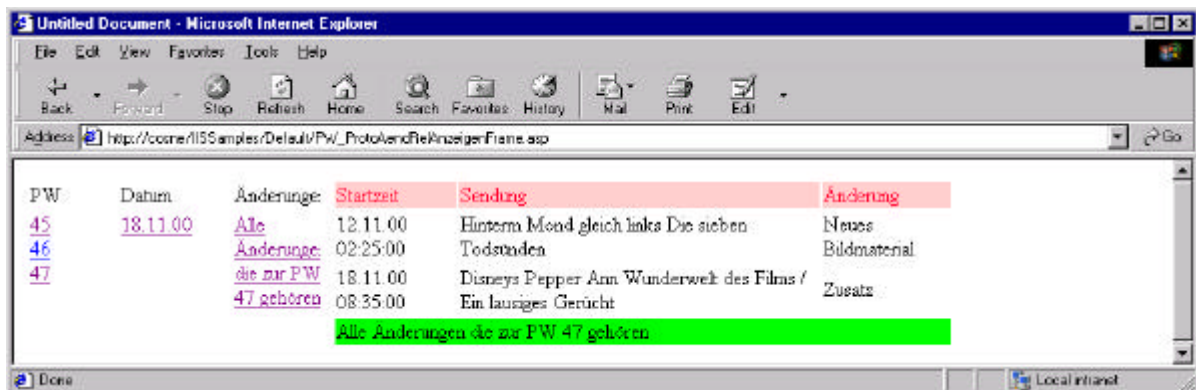


Abbildung 46 Darstellung einer redaktionellen Änderung

8.2.1.6 Änderungsmeldungen anzeigen

Diese Ansicht zeigt alle Änderungsmeldungen, sortiert nach ihrer Veröffentlichung. Der Benutzer kann sich zu jeder Meldung die dazugehörigen redaktionellen Änderungen anzeigen lassen.

8.2.1.7 Verschiedene Versionen zeigen

In dieser Ansicht kann der Benutzer die Historie der Programmablaufpläne der einzelnen Programmwochen, die gerade veröffentlicht sind, ansehen. Sortiert nach der Programmwoche (Kalenderwoche) und der Version, können die Ablaufpläne der einzelnen Sendetage abgerufen werden.

8.2.2 Material bearbeiten

Das Untermenü 'Material' gliedert sich in drei Teilbereiche auf:

- Importieren von Bildmaterial
- Zugeordnetes Material
- Restliches Material

8.2.2.1 Importieren von Bildmaterial

In diesem Dialog muß der Benutzer den Pfad angeben, an dem das Bildmaterial abgelegt ist. Zum Import wird zusätzlich noch die ASC-Datei benötigt, die über den 'Browse...'-Button ausgewählt werden kann.

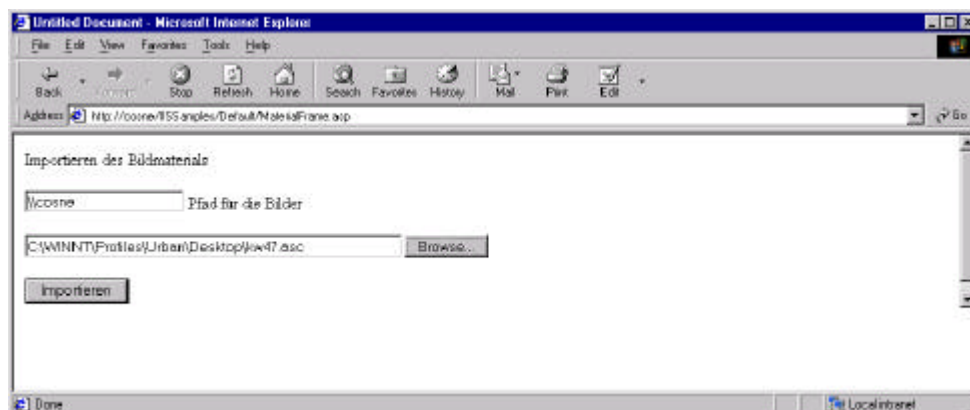


Abbildung 47 Importdialog für Material

Das Textfile wird über ein ASP-Skript ausgelesen. Die benötigten Daten - Bildname, Sendetag, Uhrzeit, Sendetitel und Bildunterschrift - werden ausgelesen und für den SQL-Aufruf bearbeitet (Hochkommatas bearbeiten). Zugleich wird überprüft, ob dieses Bild auch physikalisch auf dem angegebenen Pfad vorhanden ist. Falls ein Bild nicht gefunden wird, bricht diese Aktion mit einer Fehlermeldung ab.

Für jedes einzelne Bild werden danach die Daten an die Stored Procedure 'InsertMaterial' auf die Datenbank übergeben. Diese generiert einen eindeutigen Schlüssel für dieses Material und speichert alles in die Tabelle PI_MATERIAL. Über diese Aktion läuft eine Transaktions-

Klammer. Sollte beim Importieren des Materials ein Fehler auftreten, werden alle Bilder, die während dieses Imports schon auf der Datenbank eingetragen worden sind, wieder gelöscht. Über einen Trigger wird für jedes neu eingefügte Material eine Sendung innerhalb der Programmwoche gesucht und - falls möglich - eine Zuordnung erstellt. Über den Sendetag und die Startzeit wird der Bestandsschlüssel einer Sendung aus der Tabelle PI_PROGRAMMWOCHE herausgesucht und zusammen mit dem eindeutigen Materialschlüssel in der Tabelle PI_MATERIAL_PROGRAMMWOCHE gespeichert.

8.2.2.2 Zugeordnetes Material

Alle Materialien (Bilder, Pressemappen) mit einer Zuordnung zu einer Sendung können in diesem Unterpunkt angezeigt werden. Zugleich hat der Benutzer die Möglichkeit, falsch erkannte Zuordnungen zu lösen. Der entsprechende Eintrag wird aus der Tabelle PI_MATERIAL_PROGRAMMWOCHE gelöscht und das Material kann über den Menüpunkt 'restliches Material' einer anderen Sendung zugeordnet werden.

Sendetag	Startzeit	Sendetitel vom Material	Material	Sendetitel vom Programmwoche	Importdatum
22.04.00	13:00	Eine starke Familie	Wwz\lrich\Bild\Archiv\TestBild.jpg	HÄSCHENGESCHICHTEN	08.08.00 09:40:03

Zuordnung lösen

Abbildung 48 Darstellung des zugeordneten Materials

8.2.2.3 restliches Material

Nach Aufruf dieses Menüpunktes werden dem Benutzer alle Materialien inklusive der dazugehörigen Informationen, die noch keiner Sendung zugeordnet sind, angezeigt.

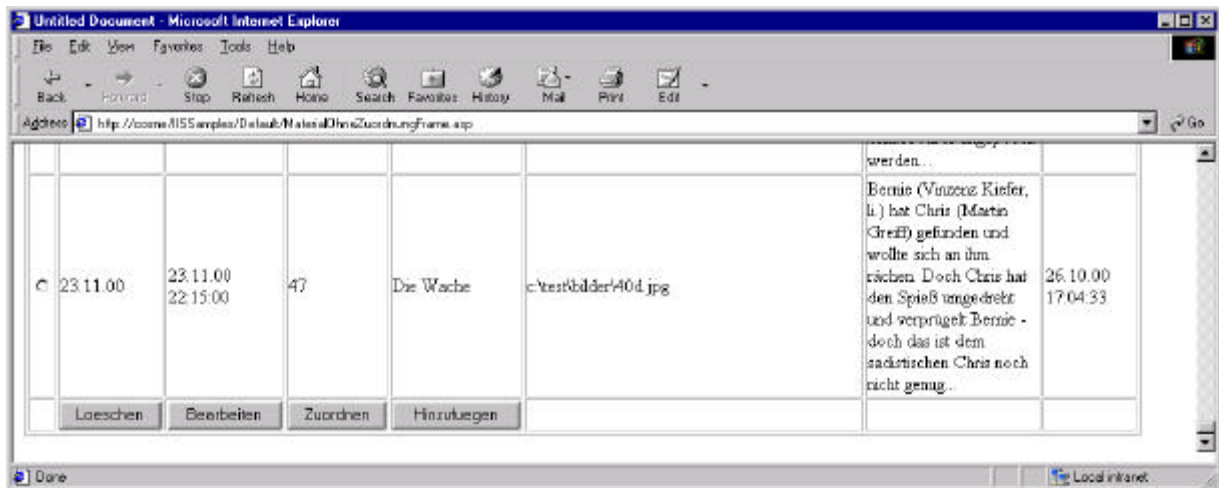


Abbildung 49 Darstellung des Materials ohne Zuordnung

Der Benutzer hat folgende Auswahl an Aktionen zum Bearbeiten des Materials:

Löschen:

Zum Löschen eines Eintrages muß ein Objekt über den Radiobutton ausgewählt und der 'Löschen'-Button gedrückt werden. Nach einer positiven Bestätigung der Sicherheitsabfrage wird das Gelöscht-Bit dieses Objektes gesetzt und ist somit für den Benutzer nicht mehr sichtbar.

Das Material wird erst von der Datenbank gelöscht, wenn die Programmwoche mit dem dazugehörigen Sendetag ebenfalls von der Datenbank gelöscht wird.

Bearbeiten:

Nach der Auswahl des Materials und betätigen des Buttons 'Bearbeiten', werden dem Benutzer alle veränderbaren Attribute angezeigt.

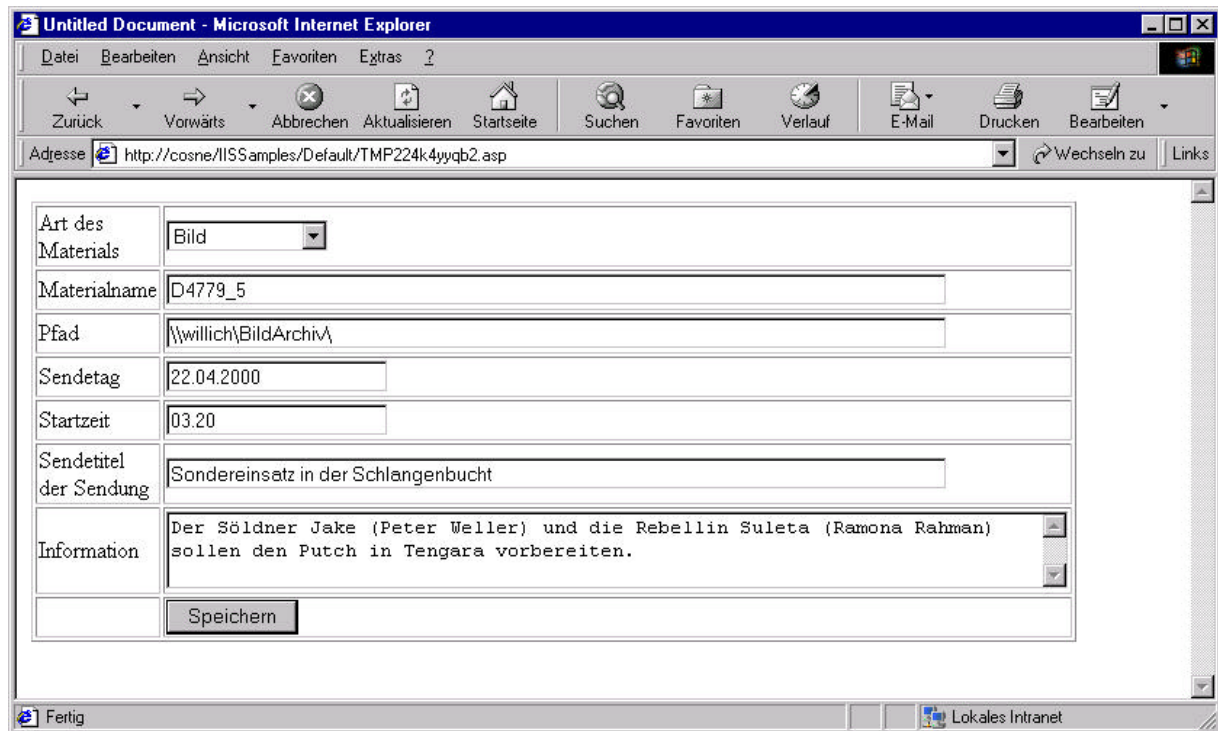


Abbildung 50 Bearbeiten der Attribute des Materials

Der Benutzer kann diese Felder nun bearbeiten und die Änderungen auf die Datenbank zurückschreiben. Dabei wird das Änderungsdatum aktualisiert.

Im Falle einer Änderung in den Feldern MAT_SENDEDATUMd oder MAT_STARTZEITd wird die alte Zuordnung - wenn vorhanden - in der Tabelle PI_MATERIAL_BESTANDSOBJEKT gelöscht, dann wird nach einer neuen passenden Sendung gesucht und gegebenenfalls die Zuordnung erstellt.

Zuordnen:

Nachdem ein Objekt ausgewählt worden ist, werden alle Sendungen des passenden Sendetages angezeigt und der Benutzer sucht sich die passende Sendung aus. Nach einer Sicherheitsabfrage wird die Zuordnung in die Tabelle PI_MATERIAL_BESTANDSOBJEKT hinzugefügt.

Hinzufügen:

Nach Aktivieren des Buttons 'Hinzufügen' wird folgender Dialog aufgerufen:

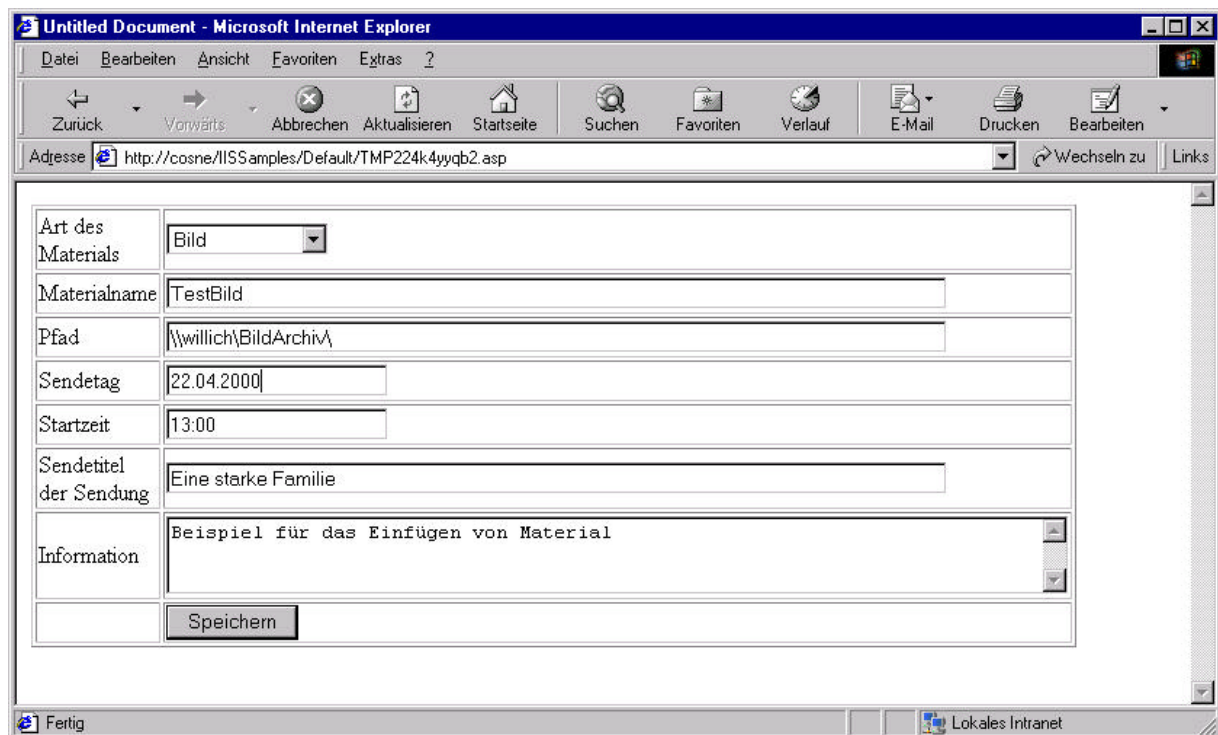


Abbildung 51 Manuelles Einfügen von Material

Folgende Felder müssen unbedingt gefüllt werden, um später eine Zuordnung zu einer Sendung finden zu können: Art des Materials, Materialpfad, Sendedatum, Startzeit und Sendetitel. Im Feld Information kann eine nähere Erläuterung (zum Beispiel eine Bildunterschrift) zu dem Material gegeben werden

Der Eintrag wird über eine Stored Procedure mit einer neuen ID in der Tabelle PI_MATERIAL gespeichert. Danach wird mit Hilfe des Sendedates und der Startzeit nach einem passenden Bestandsobjekt gesucht. Bei einer erfolgreichen Suche erfolgt ein entsprechender Eintrag in die Tabelle PI_MATERIAL_BESTANDSOBJEKT. Die Zuordnung wird dem Benutzer angezeigt.

8.2.3 Person bearbeiten

8.2.3.1 Suche nach einer Person

Nach Eingabe von Suchkriterien (bisher nur über Name und Vorname möglich) wird auf der Datenbank nach entsprechenden Einträgen gesucht. Der Benutzer erhält eine Liste von Personen, auf die das eingetragene Suchkriterium paßt.

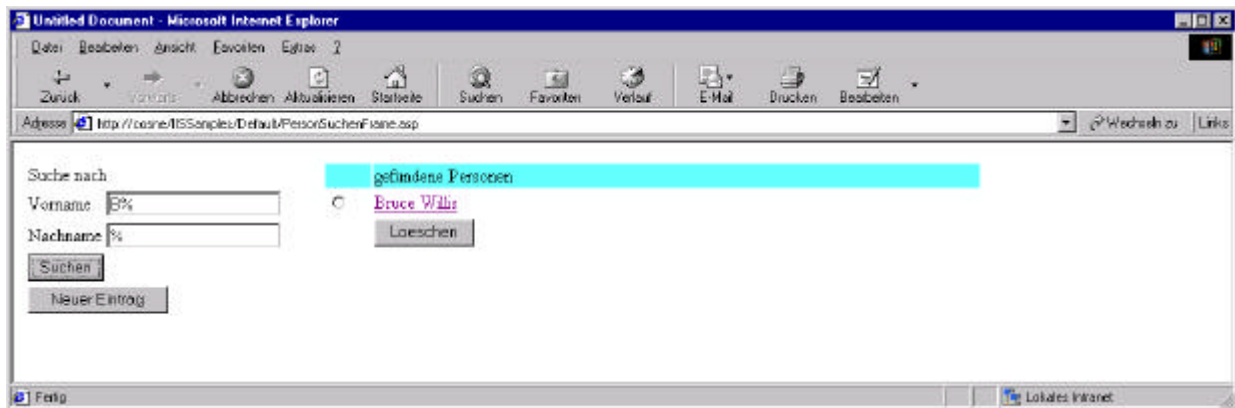


Abbildung 52 Personen suchen

8.2.3.2 Hinzufügen einer Person

Über „Neuer Eintrag“ können Personen hinzugefügt werden. In der neuen Maske müssen die bekannten Felder ausgefüllt und die Eingaben gespeichert werden.

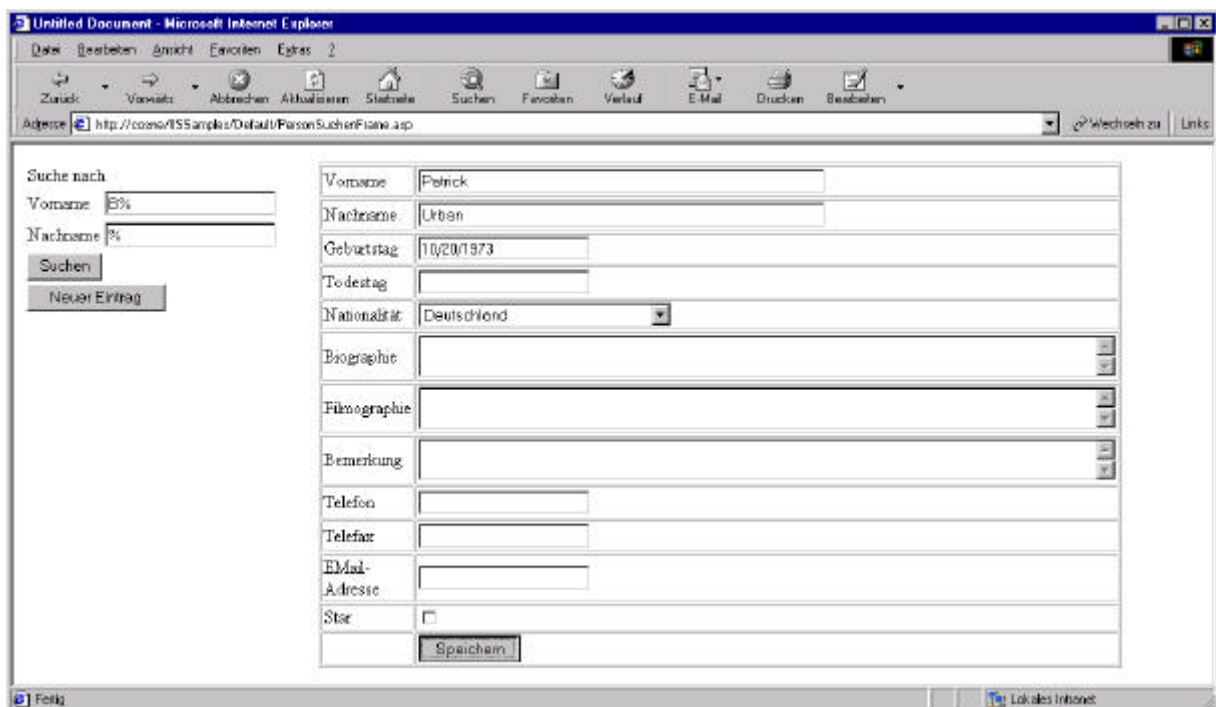


Abbildung 53 Person hinzufügen

Die neue Person erhält eine persönliche ID. Sie ist ab sofort auf der Datenbank vorhanden und kann den Sendungen zugeordnet werden.

8.2.3.3 Personendaten bearbeiten

Personen können durch Anklicken des Namens bearbeitet werden. Der Benutzer sieht alle Attribute, die zu dieser Person gehören, und kann jedes einzelne von ihnen verändern und erneut abspeichern.

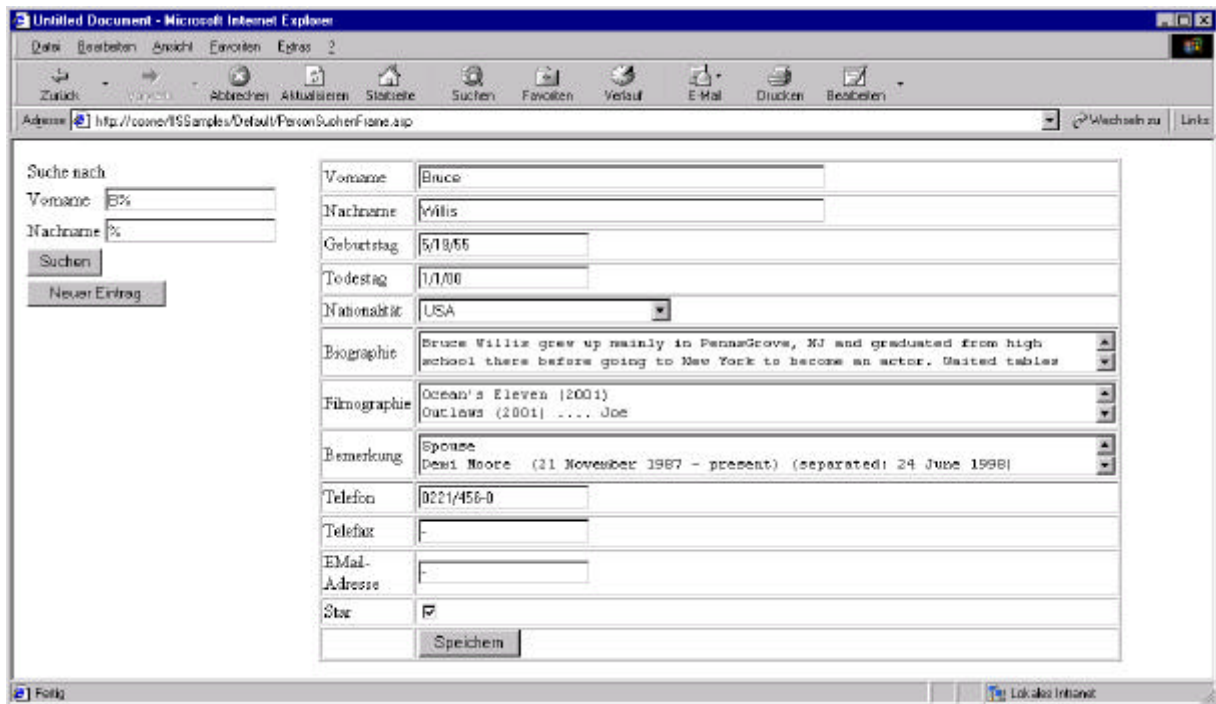


Abbildung 54 Personendaten bearbeiten

9 Zusammenfassung

Ziel der Diplomarbeit war die Konzeption und Umsetzung eines datenbankgestützten, automatisierten Systems zur Erzeugung der Programminformation eines Fernsehsenders zur Nutzung im B2B-Angebot.

Um den gesamten Kontext verstehen zu können, wurden am Anfang dieser Arbeit Grundlagen geschaffen, die für das Verständnis notwendig sind. Zum einen beziehen diese sich auf den unternehmensinternen Bereich, aber auch auf Methoden, die später in diesem Projekt verwendet wurden.

Mit erworbenen Vorwissen wurden innerhalb einer Ist-Analyse zum einen die Strukturen und Abläufe der Abteilung Presse/Programminformation aufgezeigt und im zweiten Schritt die vorhandenen Basissysteme auf ihre Verwendbarkeit als Informationsquelle für eine datenbankgestützte Programmwoche untersucht.

Aufgrund dieser Analyse wurden in der vorliegenden Diplomarbeit verschiedene Lösungsansätze aufgezeigt und letztendlich der geplante Weg zur Erstellung der Programmwoche vorgestellt.

Als nächstes wurden ein Einblick gegeben, welche Entwicklungstools für die Realisierung des Projektes verwendet wurden und mit Hilfe eines Beispiels beschrieben.

Es folgte die Realisierung der Datenbank zusammen mit der Businesslogik. Um die Funktionalität dieser zu zeigen, wurde zeitgleich ein Prototyp entwickelt, der auf diese Datenbank aufsetzt.

Zum Abschluß dieser Arbeit werden einige Erfahrungen und Probleme dieser Diplomarbeit zusammengefaßt.

Der DataArchitect hat sich als Hilfsmittel für das Datenbankdesign bewährt. Der Entwickler kann mit Hilfe der graphischen Darstellung schnell einen Überblick über das Modell gewinnen. Anpassungen und Erweiterungen der Datenbank sind schnell und einfach eingearbeitet und können direkt auf die Datenbank übertragen werden. Stored Procedures und Trigger wurden, trotz der Möglichkeit sie über DataArchitect zu generieren, manuell geschrieben. Der Grund liegt in der Verwaltung und Einspielung von Datenbankskripten. Die Procedures und Trigger, die auf eine Datenbank eingespielt werden sollen, werden in DDL-Skripten verwaltet. Innerhalb dieser Skripte gibt es Defines, mit deren Hilfe zwischen verschiedenen Datenbanksystemen unterschieden werden kann. Diese DDL-Skripte werden mit Hilfe eines C-Compilers für die jeweilige Datenbank übersetzt und eingespielt. So kann für jedes Datenbanksystem die Datenbank über ein Skript angelegt werden. Der DataArchitect

würde für jedes Datenbanksystem ein eigenes Skript generieren. Dies bedeutet mehr Arbeitsaufwand beim Erstellen der Skripte, andererseits sind diese aber einfacher zu verwalten.

Die Verwendung von Dreamweaver Ultradev zur Entwicklung des Prototyps hat sich bewährt. Ohne besondere Vorkenntnisse über HTML-Design können schnell ASP-Seiten durch Auswahl der einzelnen Objekte erzeugt werden. Der von Ultradev generierte Code ist sehr anschaulich und leicht verständlich, diente aber in den meisten Fällen nur als Grundgerüst. Zusätzlich benötigte Funktionen konnten über den integrierten Editor einfach hinzugefügt werden. Als Vorteil erwiesen sich dabei Grundkenntnisse in der Programmierung mit Visual Basic. Die Funktionen in ASP wurden in VB-Skript erstellt, welches eine Untermenge von VB ist.

Die Erstellung der Businesslogik über Stored Procedures und Trigger erwies sich als Herausforderung. Die Logik soll dem Client Arbeit abnehmen und auf die Datenbank verlagern. Deshalb mußte genau geplant werden, welche Aufgaben die Datenbank übernehmen kann und wie der Client diese Aktion ausführt. Dafür war eine enge Zusammenarbeit mit den Programmierern des endgültigen Clients notwendig. Besonders bei Stored Procedures mußten die Übergabeparameter und Rückgabewerte vorher definiert werden, damit ein reibungsloser Ablauf gewährleistet ist.

Der Client sollte keine Verbindung zu den Basissystemen (TRAFFIC/PROFIS) haben. Datenbeschaffung, Änderungsabgleich und Rückgabe der redaktionellen Texte werden ausschließlich über Stored Procedures und Trigger ausgeführt. Deshalb war es nicht möglich die Datenbank so zu realisieren, das sie auf einem separaten Server lauffähig ist. Grund dafür sind Einschränkungen, die von Sybase vorgegeben sind. Die Idee war, die benötigten Tabellen, die auf anderen Servern liegen, über Remote-Tabellen in die Programmwochendatenbank zu spiegeln. Diese haben die gleichen Eigenschaften wie ein View und unterstützen alle benötigten Funktionen. Zugleich bietet Sybase den Datentyp 'Text' an, der für alle redaktionellen Texte in den Tabellen Bestandsobjekt und Person benutzt worden ist. Im Gegensatz zum Varchar besitzt des Text-Feld keine Beschränkung auf 255 Zeichen, sondern kann unendlich lang sein. Hier zeigte sich, das dieses ein Datentyp ist, auf den über Remote-Tabellen kein Update möglich ist. Gerade bei den redaktionellen Texten ist es wichtig, daß die Änderungen an das Basissystem für die Wiederverwendbarkeit zurückgegeben werden können. Deshalb wird die Programmwochendatenbank jetzt zusammen mit PROFIS auf einem Server laufen.

Die enge Zusammenarbeit mit der Presse/Programminformation erwies sich als konstruktiv. Aufgrund ihrer Erfahrungen in der Erstellung der Programmwoche wiesen sie auf einige Details hin, die im ersten Augenblick nicht offensichtlich waren, aber trotzdem zu beachten sind und sich daher im Datenbankmodell wiederfinden. Im Gegenzug wurde durch die kontinuierliche Präsentation des Prototyps bei den Redakteuren eine Akzeptanz zum neuen Medium geschaffen.

10 Ausblick

Die Konzeption und Realisierung der Datenbank zusammen mit der Businesslogik ist mit dieser Arbeit abgeschlossen. In nächster Zeit wird den Anwendern eine Testversion des späteren Client zu Verfügung gestellt. In dieser Phase wird sich herausstellen, ob an der Datenbank noch kleinere Anpassungen gemacht werden müssen, um den kompletten Workflow zu unterstützen.

Die erste Version der Programmwoche soll im Februar zum Einsatz kommen und dann die gedruckte Version ersetzen. Das erste Release setzt die gedruckte Programminformation um. In Zukunft sollen die Möglichkeiten des Mediums Internet noch besser ausgenutzt werden. Folgende Ideen stehen zur Debatte und werden vielleicht schon im zweiten Release mit eingearbeitet:

- Personendatenbank:

Die Personendaten werden bisher nur für die Ansicht im Langablaufplan genutzt. Dabei werden nur für die Stars Biographien verwendet. Alle anderen Personen, die an einer Sendung beteiligt sind, werden nur mit Namen erwähnt. In Zukunft soll es möglich sein, Informationen zu allen Personen in der Programmwoche abzurufen. Ebenso soll die umgekehrten Abfrage gemacht werden können, in welchen Sendungen diese Person in der aktuellen Programmwoche mitwirkt.

- Links zu aktuellen Formaten:

Die Materialtabelle ist bewußt allgemein gehalten. In Zukunft soll dort nicht nur das Bildmaterial verwaltet werden, sondern auch Trailer, Filmausschnitte, Pressemappen, etc. Diese können dann Online angesehen oder auch heruntergeladen werden.

- Suchfunktion:

Eine gewünschte Erweiterung des Programmablaufplans ist eine integrierte Suchfunktion innerhalb des Programms. Der Benutzer kann eine Sendung angeben und erhält den Sendetag und die Startzeit.

- Newsletter:

Auf Wunsch können sich die Benutzer des B2B-Angebotes eines Newsletter zuschicken lassen, in der die Benutzer über alle aktuellen Änderungen und Events informiert werden.

11 Anhang

11.1 Abbildungsverzeichnis

Abbildung 1 DB-Aufrufe mit ODBC	31
Abbildung 2 Die ODBC-Architektur.....	32
Abbildung 3 Aufbau von OLE-DB.....	33
Abbildung 4 Erstellung der Programmwoche	36
Abbildung 5 Auszug aus dem Programmwochenheft für einen Film	38
Abbildung 6 Auszug aus dem Programmwochenheft für eine Serie	39
Abbildung 7 Auszug aus dem Programmwochenheft für einen Sonderablauf	39
Abbildung 8 Datenbankschnittstelle PROFIS/TRAFFIC.....	41
Abbildung 9 Ablauf zur Erstellung der neuen Programmwoche	53
Abbildung 10 Startbildschirm	56
Abbildung 11 Tabellen anlegen.....	56
Abbildung 12 Felder in Tabellen anlegen.....	57
Abbildung 13 Relationen festlegen	57
Abbildung 14 Kardinalität festlegen.....	58
Abbildung 15 Konvertieren des Modells	58
Abbildung 16 Auswahlmenü für die Generierung.....	59
Abbildung 17 Startbildschirm Ultradev	61
Abbildung 18 Datenquelle anlegen	62
Abbildung 19 Abfrage erstellen	62
Abbildung 20 Seite erstellen	63
Abbildung 21 Serververhalten festlegen.....	63
Abbildung 22 Ansicht fertige Seite	64
Abbildung 23 Programmcode im Editor	64
Abbildung 24 Datenbankmodell Programmwoche Teil 1.....	66
Abbildung 25 Datenbankmodell Programmwoche Teil 2.....	67
Abbildung 26 Startmenü.....	102
Abbildung 27 Startmenü Programmwoche bearbeiten	103
Abbildung 28 Programmablaufplan	103
Abbildung 29 Label Übersicht	104
Abbildung 30 Erstellen eines Labeltextes	104
Abbildung 31 Zuordnen eines Labels zu einer Sendung.....	105
Abbildung 32 Löschen eines Labels.....	105
Abbildung 33 Programmelement bearbeiten.....	106
Abbildung 34 Neuen Darsteller zuordnen.....	108
Abbildung 35 Auswahl einer Person	108
Abbildung 36 Darstellerliste	109
Abbildung 37 Stab zuordnen.....	109
Abbildung 38 Hinzufügen eines Programmwocheneintrages.....	110
Abbildung 39 Manuell eingefügtes Programmwochenelement.....	110
Abbildung 40 Löschen eines Programmwocheneintrages.....	111
Abbildung 41 Übersicht aller elementaren Änderungen	113
Abbildung 42 Attribute eine Änderung bearbeiten.....	114
Abbildung 43 Änderung manuell hinzufügen	114
Abbildung 44 Auswahl einer Sendung	115
Abbildung 45 redaktionelle Änderung erstellen.....	116
Abbildung 46 Darstellung einer redaktionellen Änderung.....	116
Abbildung 47 Importdialog für Material.....	117
Abbildung 48 Darstellung des zugeordneten Materials	118
Abbildung 49 Darstellung des Materials ohne Zuordnung	119
Abbildung 50 Bearbeiten der Attribute des Materials	120
Abbildung 51 Manuelles Einfügen von Material	121
Abbildung 52 Personen suchen	122
Abbildung 53 Person hinzufügen	122
Abbildung 54 Personendaten bearbeiten.....	123

11.2 Literaturverzeichnis und Quellennachweis

[1]	Dr. Norbert Ulamec Datenbank Grundlagen Adresse: http://www.fh-sbg.ac.at/~ulamec/db_grund/
[2]	Internet.com Corporation Interactive-On-line SQL Tutorial Adresse: http://www.sqlcourse.com/intro.html
[3]	Tim Ehlert Sybase Architektur Adresse: http://www.iief.de/people/tim/Sybase/intro/#Transact%20SQL
[4]	Stefan Falz ASP Datenbankkurs Adresse: http://www.asp-database.de/aspdatabase/ASP_Allgemein.asp
[5]	INAS Datentechnik GmbH Was ist ODBC? Adresse: http://www.inas.de/qbref/QBase294.htm
[6]	E. Bablick, E. Obermeier, A. Roth, Ch. Zeitler Offene Systeme Adresse: http://rfhnt8001.fh-regensburg.de/dvt97/os/os.html
[7]	Uwe Logemann Scripting Adresse: http://www.w3com.de/uwe.logemann/html/scripting.asp
[8]	Sybase, Inc. Stored Procedure Processing and Management Adresse: http://my.sybase.com/detail?id=1323
[9]	Sybase, Inc. Stored Procedure Processing and Management Adresse: http://my.sybase.com/detail?id=1217
[10]	Sybase, Inc. Sybase Adaptive Server Enterprise Performance and Tuning Guide © 1997 Sybase, Inc. Document ID: 32645-01-1150-02
[11]	Microsoft Corporation Schreiben von ASP-Skripten Adresse: http://www.altmuehl.net.de/~schneider-m/ASP_Basics/Tutorial/guide/asgwri.htm
[12]	Stefan M. Lang, Peter C. Lockmann Datenbankeinsatz © 1995 Springer Verlag ISBN: 3540585583
[13]	Rene Steiner Theorie und Praxis relationaler Datenbanken ©1999 Vieweg Verlag ISBN: 3528254270
[14]	Sanjaya Hettihewa Active Server Pages in 14 Tagen ©1998 Markt und Technik Buch- und Software Verlag GmbH ISBN: 3-8272-2018-1

[15]	<p>Sybase, Inc. Sybase Adaptive Server Enterprise Reference Manual: Volume 1: Commands and Functions © 1997 Sybase, Inc. Document ID: 32401-01-1150</p>
[16]	<p>Sybase, Inc. Sybase Adaptive Server Enterprise Reference Manual: Volume 2: Procedures © 1997 Sybase, Inc. Document ID: 32402-01-1150</p>
[17]	<p>Sybase, Inc. Sybase Adaptive Server Enterprise Reference Manual: Volume 3: Data Types and System Tables © 1997 Sybase, Inc. Document ID: 32416-01-1150</p>
[18]	<p>Onlinehilfe Dreamweaver Ultradev</p>
[19]	<p>Onlinehilfe PowerDesigner 6.1</p>

11.3 Abkürzungsverzeichnis

ADO	ActiveX Data Objects
API	Application Programming Interface
ASE	Adaptive Server Enterprise
ASP	Active Server Pages
B2B	Business-to-Business
CGI	Common Gateway Interface
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
DBMS	Database Management System
FSK	freiwillige Selbstkontrolle
FTP	File Transfer Protocol
HTTP	Hyper Text Markup Language
ODBC	Open Database Connectivity
OLE-DB	Objekt Linking and Embedding - Database
OLTP	On-Line Transaction Processing
PROFIS	Program - und Fil minformationssystem
PW	Programmwoche (entspricht einer Woche, nur von Samstag bis zum darauffolgenden Freitag)
SAP	Systeme, Anwendungen, Produkte
SQL	Structured Query Language
Tag	Formatierungsanweisung
XML	Extensible Markup Language

11.4 Glossar

Applikation	Anwendung, die ausgeführt werden kann
Browser	Software zum Suchen und Darstellen von Webseiten
B2B	Einsatz von Datenverarbeitung und Telekommunikation, um Geschäfte zwischen Unternehmen zu unterstützen
Businesslogik	Funktionen zur Unterstützung des Workflows
Button	Schaltflächen, die Aktionen innerhalb einer Anwendung auslösen
Checkbox	Schalter, um Optionen ein-/ auszuschalten oder mehrere Objekte zu markieren
Client	Anwendung beim User
Data-Warehouse	sehr groß Datensammlung, welche meist aus verschiedenen, großen Datenbanken bestehen
FTP	File Transfer Protocol: ermöglicht die Übertragung/ Transfer von Dateien zwischen verschiedenen Computern über das Netz
Internet	Internationales Netzwerk auf TCP/IP-Basis. Es besteht aus mehreren Diensten, darunter ftp, E-Mail, Web und Telnet.
Intranet	Unternehmens-Netzwerk auf TCP/IP-Basis
Link	direkt ausführbarer Verweis auf ein anderes Dokument, häufig im Internet zu finden.
OLE-DB	eine Datenbankarchitektur, die unabhängig vom Datentyp, über das Netzwerk eines Unternehmens eine universelle Datenintegration (vom Großrechner bis zum Desktop) bietet.
Pfad	Verzeichnisstruktur auf dem Computer, um Dateien wiederzufinden
Proxy	wörtl. "Handlungsbevollmächtigter". Proxies sind Rechner, die in lokale Netzwerke zwischengeschaltet werden. Sie können die Geschwindigkeit steigern, indem sie als Cache eingesetzt werden und vom Rechner oft nachgefragte Pakete nicht jedesmal neu anfordern.
Radiobutton	Schalter, eine Option ein-/ auszuschalten oder ein Objekt zu markieren
ReRun	für einen Fernsehsender kostenloses Recht eine Sendung innerhalb von 24 Stunden nocheinmal auszustrahlen
Redaktionssystem	Programm zur Erstellung und Verwaltung von Web-Seiten
reverse-engineering	Wiederherstellungprozess eines Designs durch die Evaluierung des Endproduktes

Server	ein Computer, der anderen Computer bestimmte Dienste zur Verfügung stellt
Template	Formatvorlage zur Generierung von statischen und dynamischen Internetseiten

11.7

11.5 Inhalt der CD-ROM

Die CD-ROM, die dieser Diplomarbeit bei liegt, enthält folgende Verzeichnisse:

- Diplomarbeit:
Text der kompletten Diplomarbeit in Word-Dokumenten
- Datenbank:
alle Skripte zur Generierung der Datenbank untergliedert in:
- Tabellen:
Jede Tabelle der Programmwoche kann bis zu 6 Dateien aufteilen, die sich wie folgt untergliedern:
 - *Tabellenname.tab*: Tabellendefinition
 - *Tabellenname.trg*: Trigger, die auf dieser Tabelle liegen
 - *Tabellenname.ins*: vordefinierte Einträge für die Tabelle
 - *Tabellenname.idx*: Indexe, die auf dieser Tabelle liegen
- *Tabellenname.per*: Rechte der Gruppen, die auf diese Tabelle zugreifen dürfen
 - *Tabellenname.key*: Primär- und Fremdschlüssel der Tabelle
- Stored Procedures:
Alle Stored Procedures sind in der Datei *PWWoche.syb* zusammengefaßt.
- Views:
Alle Views sind in der Datei *PWViews.syb* zusammengefaßt.
- Header-Dateien:
Die zuvor genannten Datenbankskripte werden über einen C-Compiler generiert. Die Headerdateien beinhalten Defines, die das Umstellen der Datenbank und Ändern von Tabellen vereinfachen sollen. Folgende Dateien befinden sich in diesem Verzeichnis:
 - *PWWOCH.h*: Defines, die nur die Programmwoche betreffen
 - *dbconfig.h*: welcher Kunde benutzt welchen Server
 - *dbserver.h*: Syntaxbestimmung zwischen SQL und Sybase-Server
- ASP:
Alle ASP-Dateien für den Prototypen

11.6 Erstellte Stored Procedures

11.7 Trigger auf den Tabellen