

Softwarepraktikum 2003

Buchführungssystem

Projektdokumentation
(Teilbeleg 1)

Teamleiter: Dominique Petersen

Mitglieder des Projektteams:

Annette Keusch
Nadine Pollmer
Dirk Brückner
Marco Döring

Praktikumsbetreuer: Christian Meichsner

Chemnitz, den 11.05.2003

Inhaltsverzeichnis

1	Spezifikation der funktionellen Anforderungen	3
1.1	Produktbeschreibung	3
1.2	Funktionelle Anforderungen	4
1.2.1	Umgebungsmodell	4
1.2.2	Verhaltensmodell	6
1.3	Definition der Nutzerschnittstelle	32
2	Spezifikation der operationellen Anforderungen	50
2.1	Operationelle Anforderungen an die Daten und die Datenbasen.....	50
2.2	Operationelle Anforderungen an die Datenströme	53
2.3	Operationelle Anforderungen an die Prozesse.....	53
3	Spezifikation wichtiger Qualitätsanforderungen	54
3.1	Benutzbarkeit	54
3.2	Zuverlässigkeit.....	54
3.3	Integrität	54
3.4	Flexibilität	54
3.5	Portabilität.....	55
4	Basismaschine und Entwicklungsumgebung	56
4.1	Basismaschine.....	56
4.1.1	Hardwarekonfiguration	56
4.1.2	Betriebssystem	56
4.1.3	Sonstige Basissoftware	56
4.2	Entwicklungsumgebung.....	57
5	Planung	58

1 Spezifikation der funktionellen Anforderungen

1.1 Produktbeschreibung

Das Buchführungssystem (BFS) unterstützt ein Handelsunternehmen kleinerer bis mittlerer Größe bei der Durchführung sämtlicher buchhalterischer Aufgaben. Das System verwaltet gemäß der doppelten betrieblichen Buchführung ein Kontensystem, welches die verschiedenen Mittelherkünfte (Passiva), Mittelverwendungen (Aktiva) sowie Aufwendungen und Erträge abbildet und schließlich den jährlichen Kontenabschluss und die jährliche Bilanzerstellung unterstützt. Desweiteren wird ein Grundbuchsystem geführt und die Buchung von Geschäftsfällen durch gegebene Buchungssätze (Soll an Haben) ermöglicht. Aufgrund gesetzlicher Bestimmungen werden sämtliche Datenbestände mindestens 10 Jahre lang gespeichert.

Nutzungsumgebung: Das System soll auf einem einzelnen Server betrieben werden, der vorzugsweise nur berechtigten Personen zugänglich ist. Der Zugriff auf das System erfolgt durch Clients mittels Webbrowser. Ein zusätzliches Backup der Systemdaten, etwa durchgeführt durch die EDV-Abteilung des Unternehmens, wird empfohlen.

Nutzergruppen: Bei den Nutzern des Systems handelt es sich ausschließlich um berechnete Sachbearbeiter (= **Administratoren**), die über sämtliche Nutzungsrechte verfügen. Deshalb erfolgt neben der Gruppe der Administratoren keine Einteilung in weitere Nutzerklassen.

Eine genaue Definition der Rechte ist auch aus der nachfolgenden Ereignistabelle entnehmbar.

1.2 Funktionelle Anforderungen

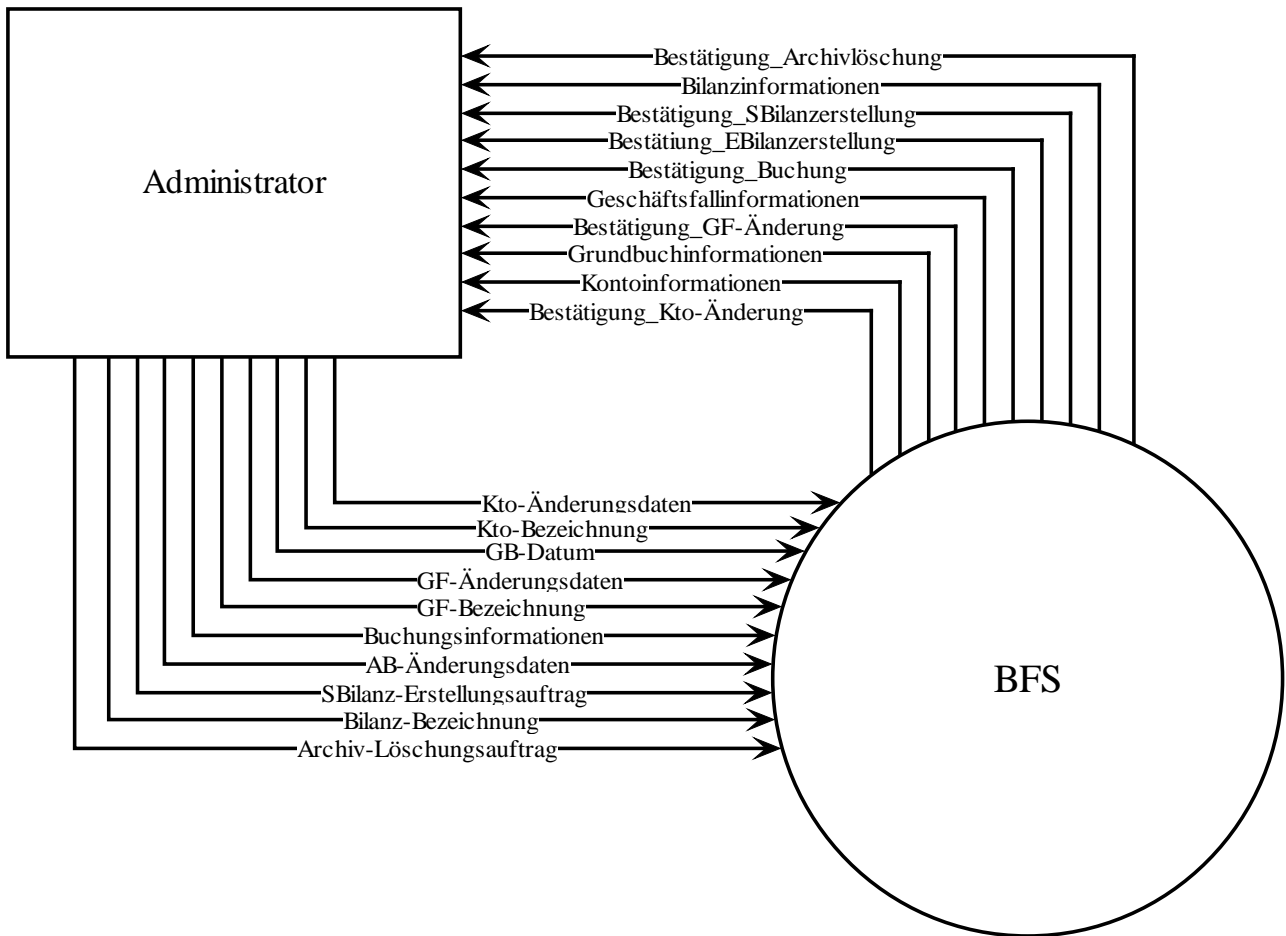
1.2.1 Umgebungsmodell

1.2.1.1 Ereignistabelle

(alle Ereignisse vom Administrator hervorgerufen)

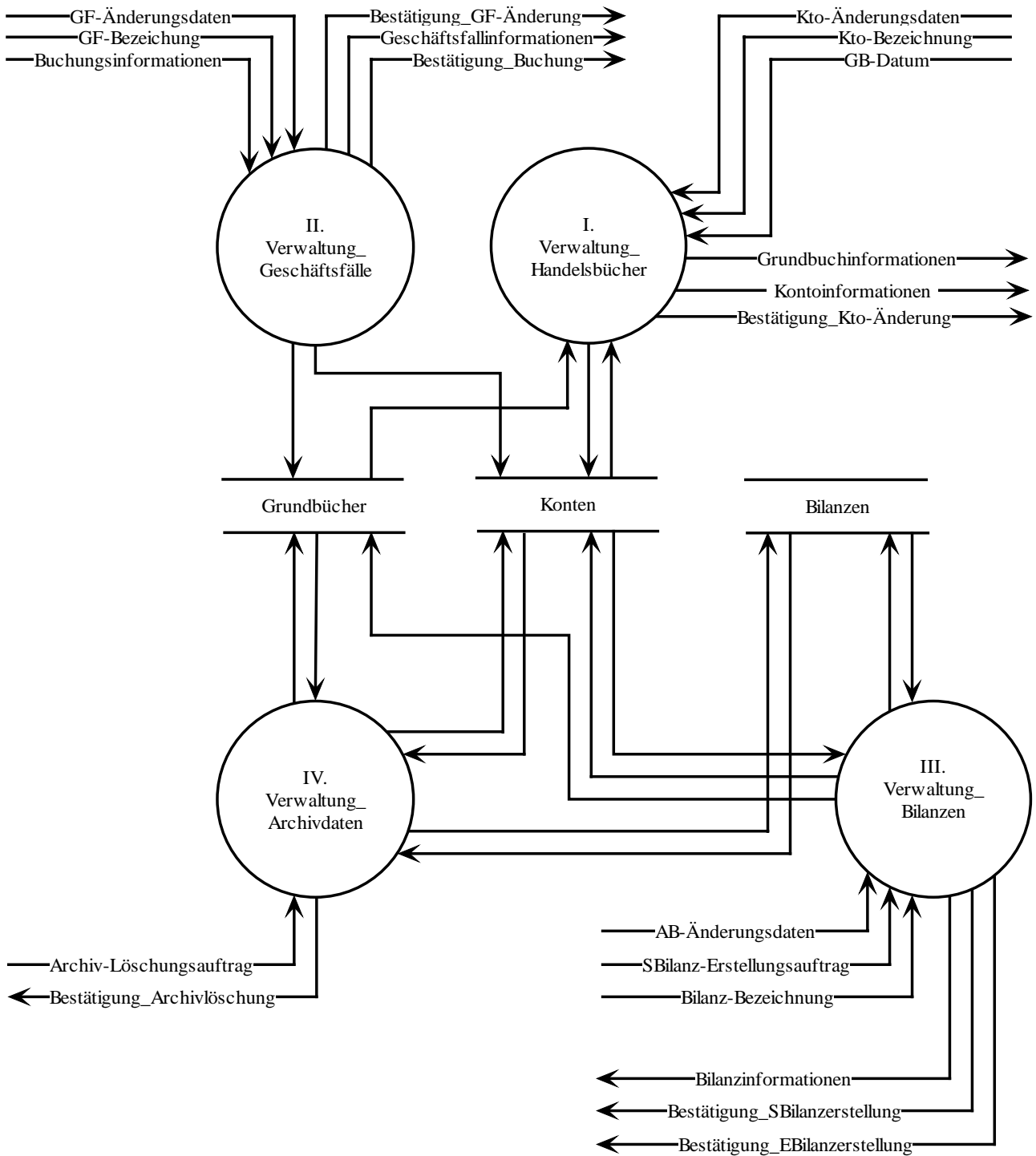
Nr.	Ereignis	Datenfluss zum System	Antwort des Systems
1	Konto anlegen	Kto-Änderungsdaten	Bestätigung_Kto-Änderung
2	Konto anzeigen	Kto-Bezeichnung	Kontoinformationen
3	Grundbuch anzeigen	GB-Datum	Grundbuchinformationen
4	Geschäftsfall anlegen oder löschen	GF-Änderungsdaten	Bestätigung_GF-Änderung
5	Geschäftsfall anzeigen	GF-Bezeichnung	Geschäftsfallinformationen
6	Buchung durchführen	Buchungsinformationen	Bestätigung_Buchung
7	Eröffnungsbilanz mittels Inventarliste bzw. AB-Korrektur erstellen (Konteneröffnung)	AB-Änderungsdaten	Bestätigung_EBilanzerstellung
8	Schlussbilanz erstellen (Kontenabschluss)	SBilanz-Erstellungsauftrag	Bestätigung_SBilanzerstellung
9	Bilanz anzeigen	Bilanz-Bezeichnung	Bilanzinformationen
10	Archivdaten löschen (löscht alle Datenbestände aus Konten, Bilanzen und Grundbüchern, die älter als 10 Jahre sind)	Archiv-Löschungsauftrag	Bestätigung_Archivlöschung

1.2.1.2 Kontextdiagramm



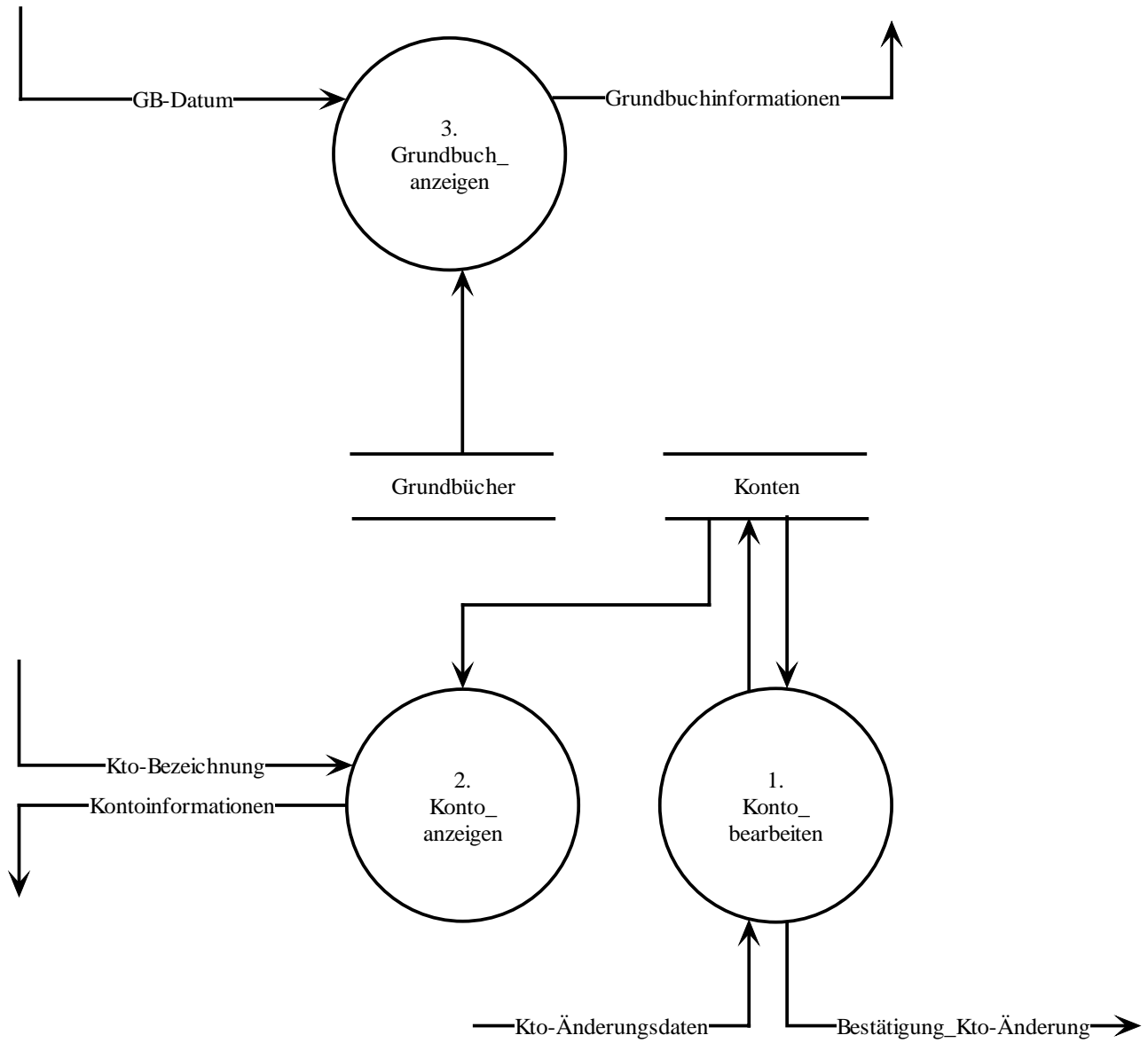
1.2.2 Verhaltensmodell

1.2.2.1 Grobes Verhaltensmodell (vergrößertes primäres Verhaltensmodell)

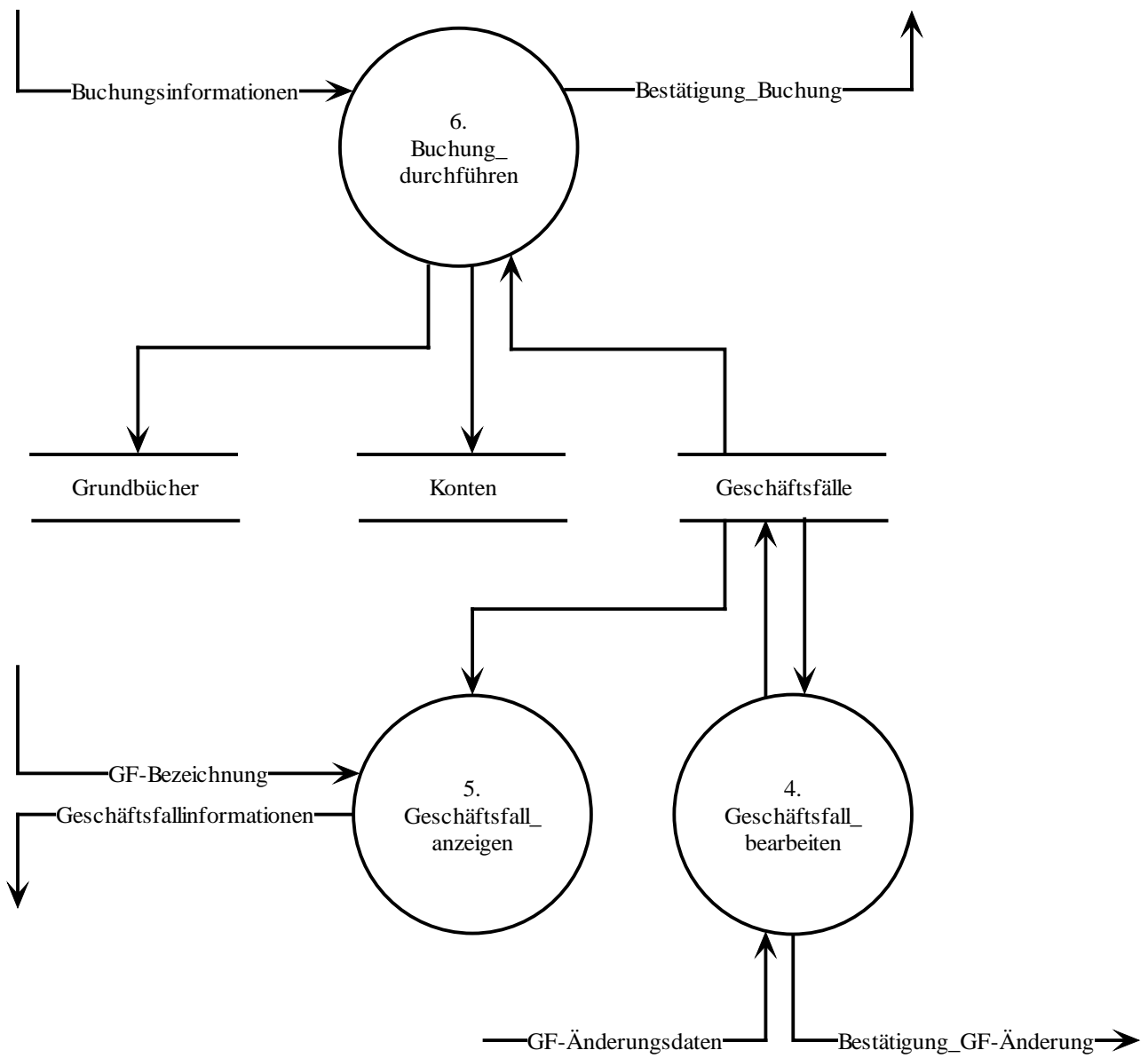


1.2.2.2 Primäres Verhaltensmodell

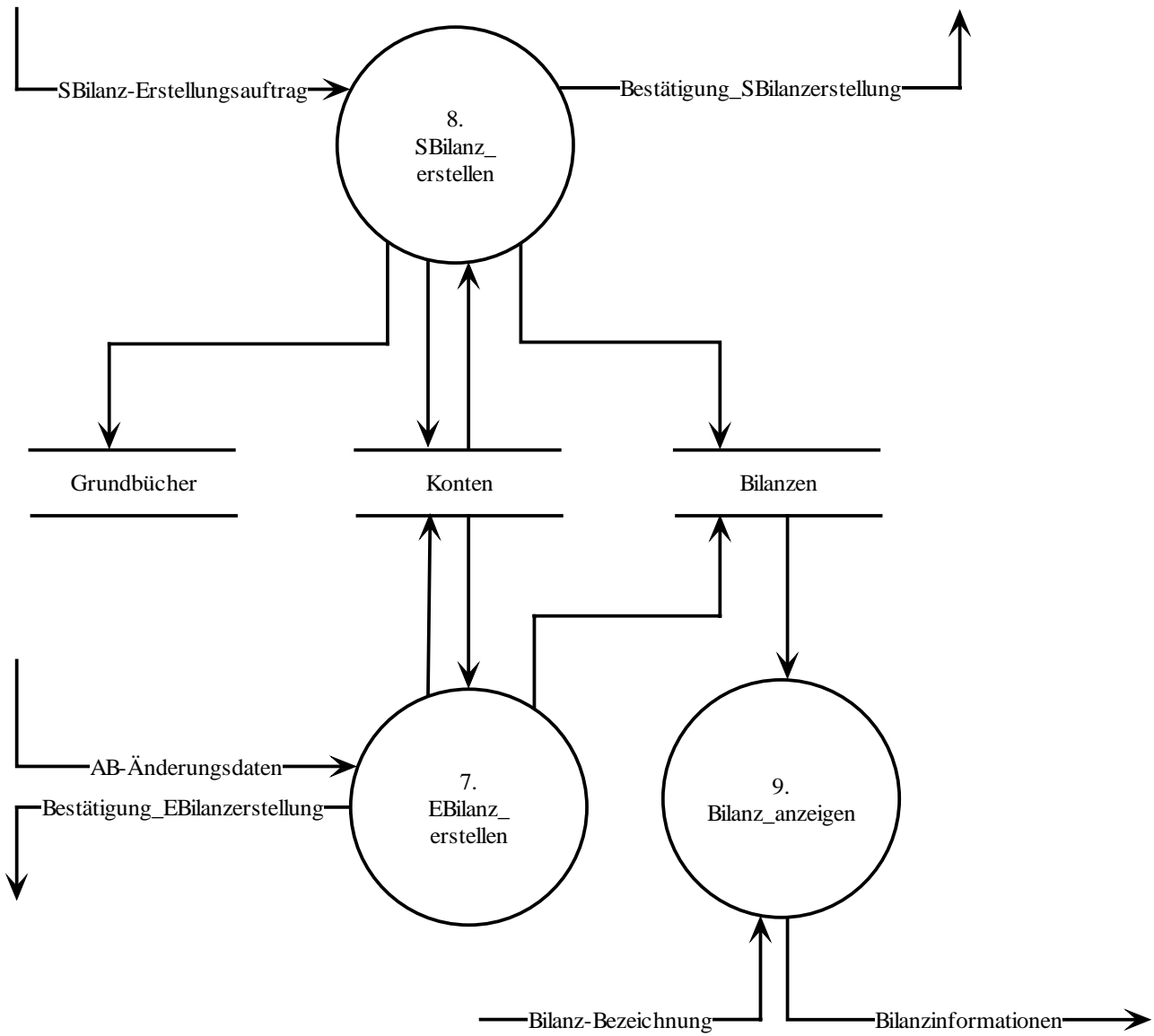
1.2.2.2.1 Teilmodell I: Verwaltung_Handelsbücher



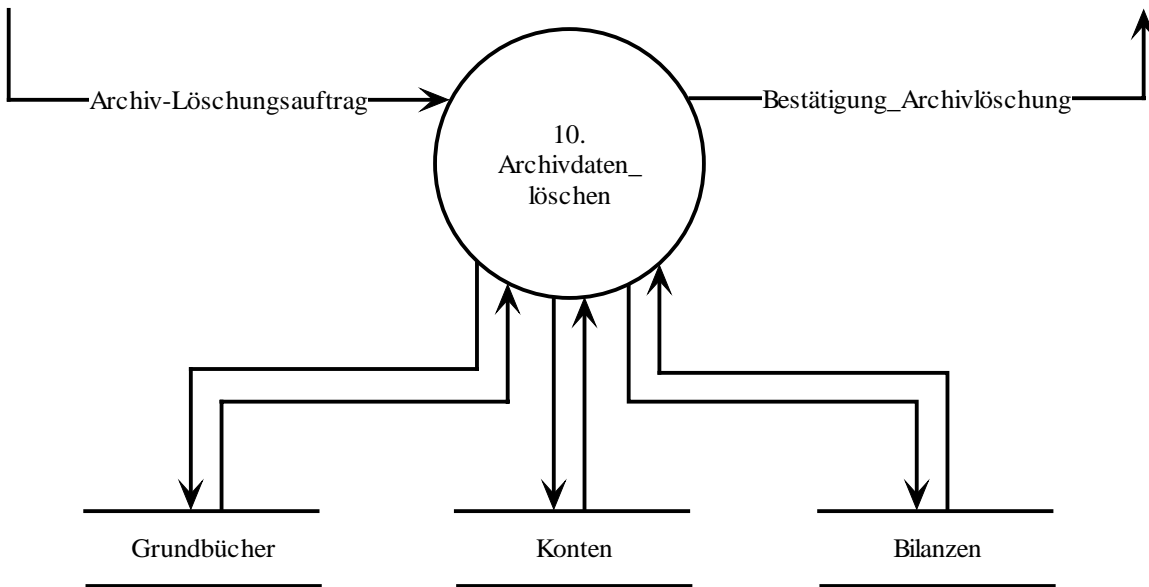
1.2.2.2.2 Teilmodell II: Verwaltung_Geschäftsfälle



1.2.2.2.3 Teilmodell III: Verwaltung_Bilanzen

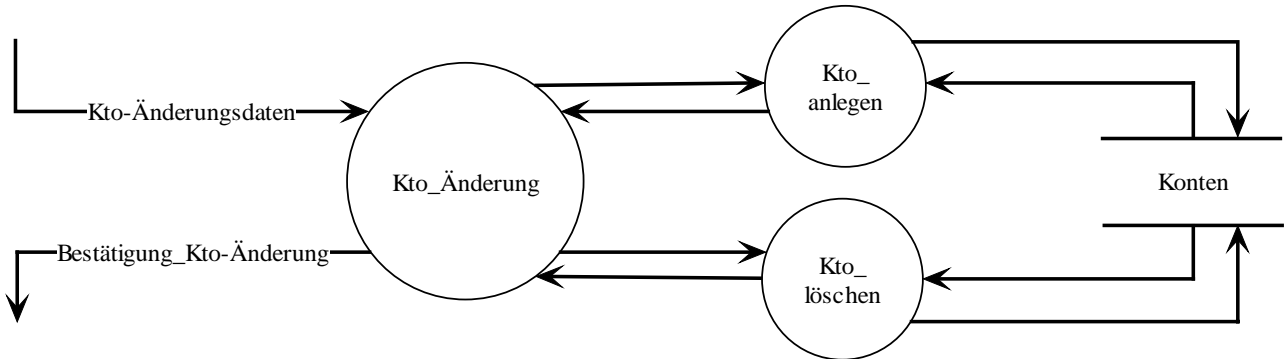


1.2.2.2.4 Teilmodell IV: Verwaltung_Archivdaten

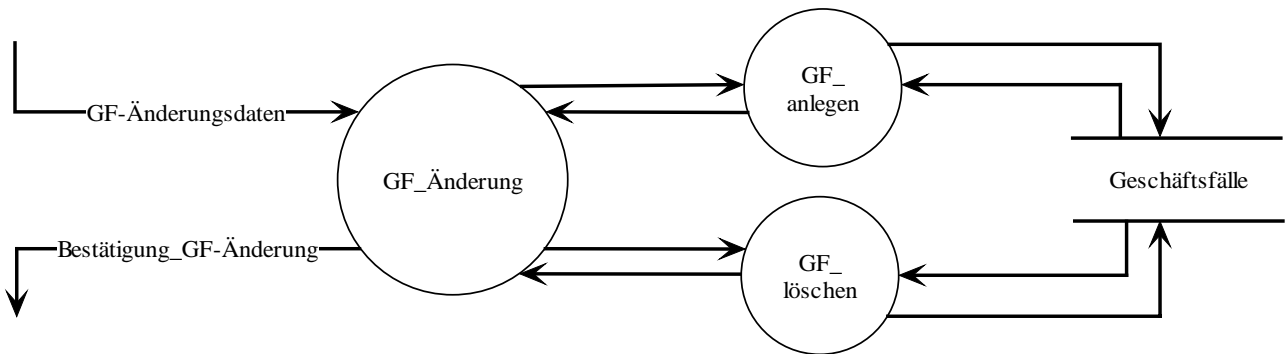


1.2.2.3 Verfeinerung der Prozesse des primären Verhaltensmodells

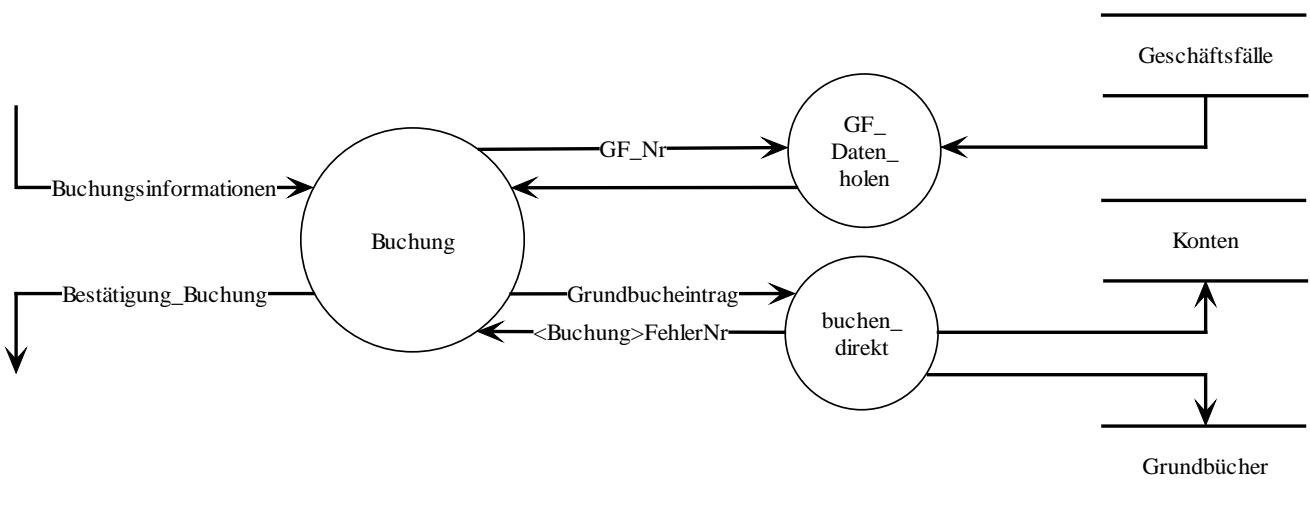
1.2.2.3.1 Verfeinerung zum Prozess: Konto_bearbeiten



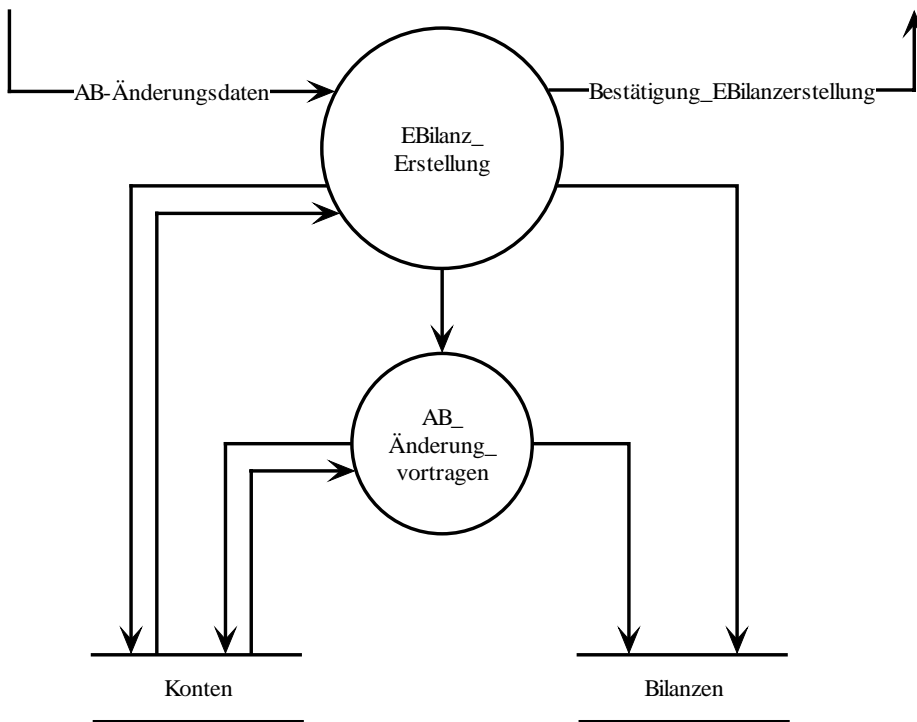
1.2.2.3.2 Verfeinerung zum Prozess: Geschäftsfall_bearbeiten



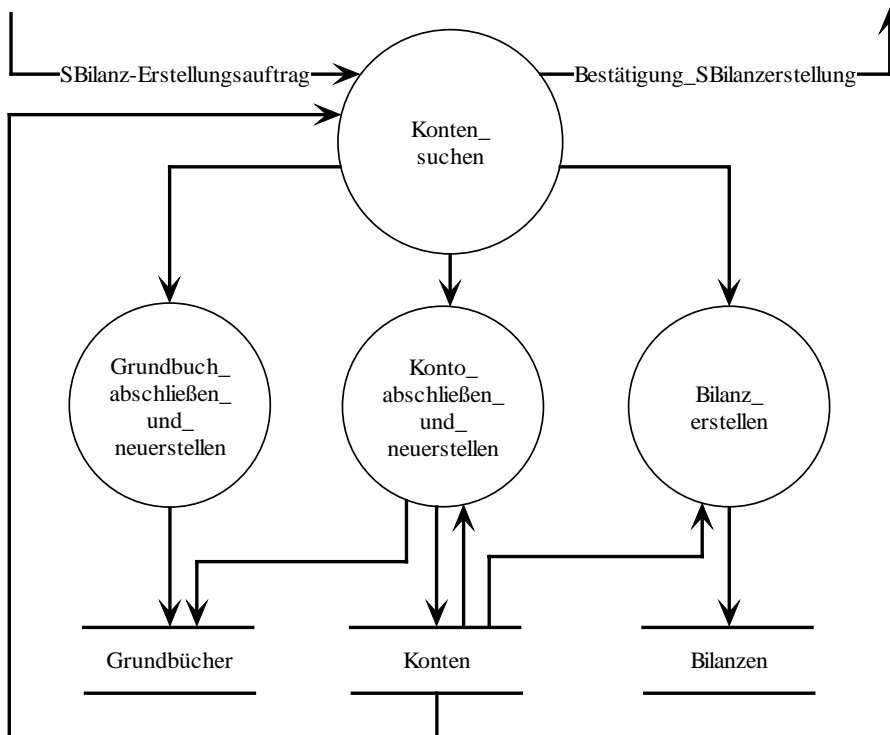
1.2.2.3.3 Verfeinerung zum Prozess: Buchung_durchführen



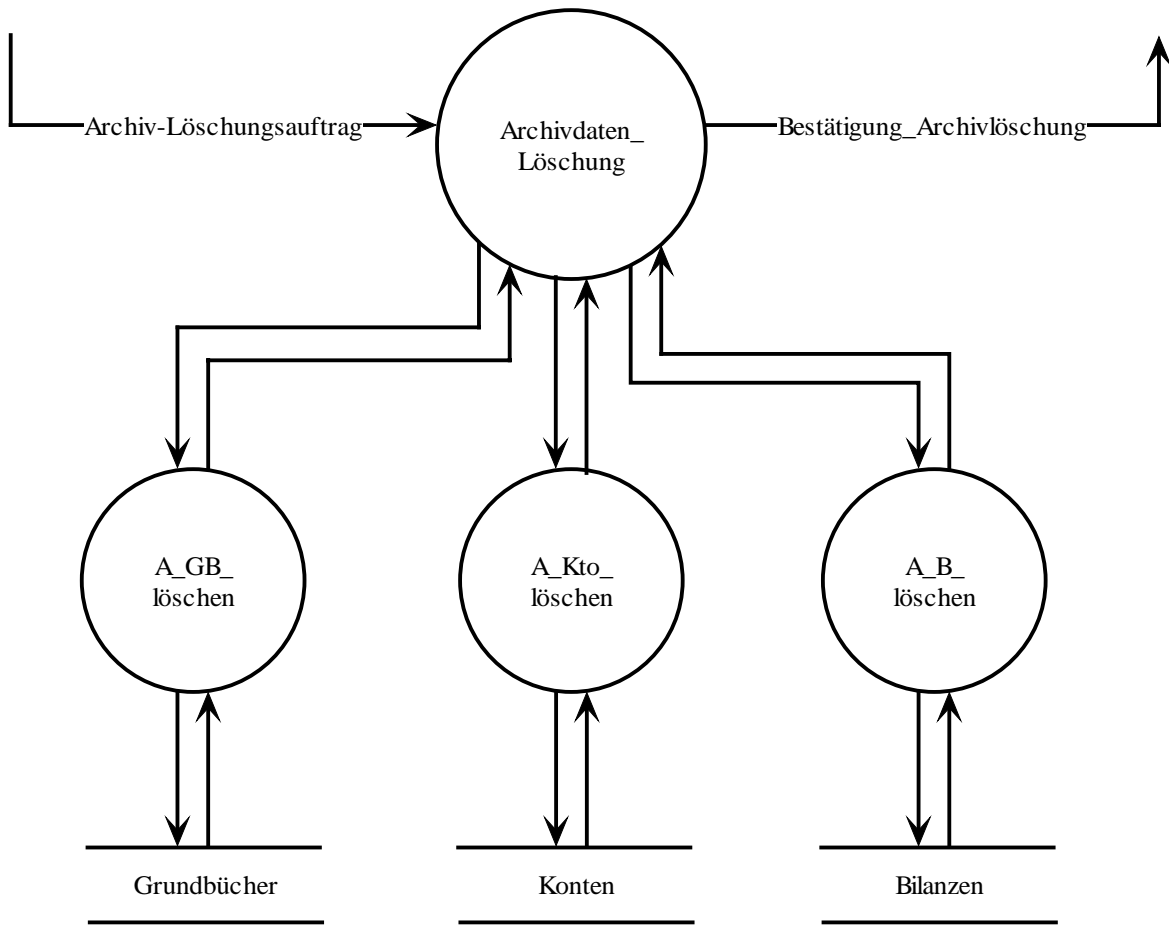
1.2.2.3.4 Verfeinerung zum Prozess: EBilanz_erstellen



1.2.2.3.5 Verfeinerung zum Prozess: SBilanz_erstellen



1.2.2.3.6 Verfeinerung zum Prozess: Archivdaten_löschen

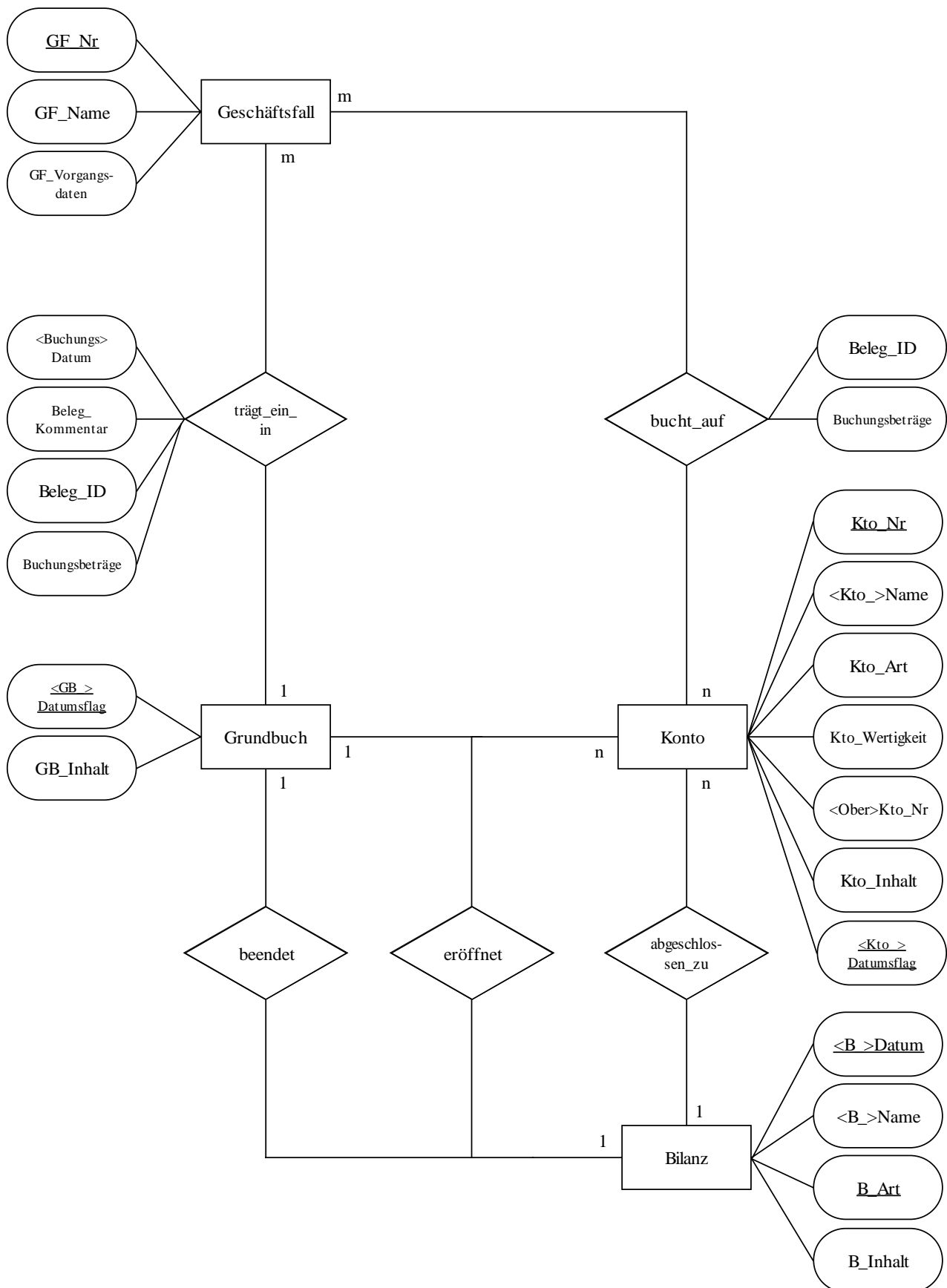


1.2.2.4 Datenkatalog

Element	Strukturbeschreibung
Konten	{Konto}
Konto	@Kto_Nr + <Kto_>Name + Kto_Art + @<Kto_>Datumsflag + Kto_Wertigkeit + <Ober>Kto_Nr + Kto_Inhalt * SBK hat die vom System vergebene <Ober>Kto_Nr = 0000 *
Kto_Nr	gZahl * 4stellig *
Kto_Art	["aktiv" "passiv" "SBK" "EK"]
Kto_Wertigkeit	gZahl * 1stellig, 0 <= Kto_Wertigkeit ; SBK hat Kto_Wertigkeit = 0, Hauptkonten 1, usw. *
Kto_Inhalt	<Soll_>Kto_Seite + <Haben_>Kto_Seite
Kto_Seite	Buchungseinträge + <Kto_Seiten>Summe
Buchungseinträge	{Buchungseintrag}
Buchungseintrag	[(Beleg_ID) (<Gegen>Kto_Nr) + <GegenKto_>Name] + <Gegen_>Betrag * während des Geschäftsjahres wird Beleg_ID verwendet, <Gegen>Kto_Nr und <GegenKto_>Name nur bei Abschlussbuchungen und beim Vortragen des AB (dann <GegenKto_>Name = "AB" und <Gegen>Kto_Nr = 0000) *
Grundbücher	{Grundbuch}
Grundbuch	@<GB_>Datumsflag + GB_Inhalt
GB_Inhalt	{Grundbucheintrag}
Grundbucheintrag	<Buchungs>Datum + Beleg_ID + Beleg_Kommentar + Kontenliste
Beleg_ID	Datum + Zeit
Beleg_Kommentar	Zeichenkette80
Kontenliste	2{Kontenlisteneintrag}
Kontenlisteneintrag	Buchungssatzzeile_blanko + <Buchungs>Betrag
Buchungssatzzeile_blanko	Kto_Nr + <Kto_>Name + betr_Kto_Seite
betr_Kto_Seite	["Soll" "Haben"]
Bilanzen	{Bilanz}
Bilanz	@<B_>Datum + <B_>Name + @B_Art + B_Inhalt
B_Art	["SB" "EB"]
B_Inhalt	<Aktiva_>B_Seite + <Passiva_>B_Seite
B_Seite	B_Seiteninhalt + <B_Seiten>Summe
B_Seiteninhalt	{B_Posten}
B_Posten	BP_Nr + BP_Name + <BP_>Betrag
BP_Nr	Kto_Nr * Konten mit Kto_Wertigkeit = 1 treten in der Bilanz auf *
BP_Name	<Kto_>Name * Konten mit Kto_Wertigkeit = 1 treten in der Bilanz auf *
Geschäftsfälle	{Geschäftsfall}
Geschäftsfall	@GF_Nr + <GF_>Name + GF_Vorgangsdaten
GF_Nr	gZahl * 3stellig *
GF_Vorgangsdaten	2{Buchungssatzzeile_blanko}
Datumsflag	["-1" "0" <Abschluss>Datum] * Datumsflag = "-1" nur für Konten während des Kontenabschlussvorgangs; Datumsflag = "0" bei laufenden Konten und laufendem Grundbuch *
Summe	Betrag
Betrag	rZahl * 12stellig, davon 2 nach Komma + Vorzeichen; in EUR *
Datum	Tag + Monat + Jahr
Tag	gZahl * 2stellig, 01 <= Tag <= 31 *
Monat	gZahl * 2stellig, 01 <= Monat <= 12 *
Jahr	gZahl * 4stellig, 2000 <= Jahr <= 2100 *
Zeit	Stunde + Minute + Sekunde
Stunde	gZahl * 2stellig, 00 <= Stunde <= 23 *
Minute	gZahl * 2stellig, 00 <= Minute <= 59 *
Sekunde	gZahl * 2stellig, 00 <= Sekunde <= 59 *
Name	Zeichenkette80

Element	Strukturbeschreibung
benannte DFD-Datenflüsse	
Kto-Änderungsdaten	["Konto anlegen" + Kto_Nr + <Kto_>Name + Kto_Art + (<Ober>Kto_Nr)] * optionale Teile der 1. Option entfallen nur beim SBK und bei den Hauptkonten *
Kto-Änderungsauswahl	["Konto anlegen"]
Bestätigung_Kto-Änderung	[Bestätigung_Kto_anlegen]
Bestätigung_Kto_anlegen	<Kto_anlegen>FehlerNr + (Konto) * optionaler Teil nur bei FehlerNr = 0 *
Kto-Bezeichnung	<Kto_>Datumsflag + [Kto_Nr <Kto_>Name]
Kontoinformationen	Konto
GB-Datum	<GB_>Datumsflag
Grundbuchinformationen	Grundbuch
GF-Änderungsdaten	["Geschäftsfall anlegen" + Geschäftsfall "Geschäftsfall löschen" + GF_Nr + (<GF_>Name)]
GF-Änderungsauswahl	["Geschäftsfall anlegen" "Geschäftsfall löschen"]
Bestätigung_GF-Änderung	[Bestätigung_GF_anlegen Bestätigung_GF_löschen]
Bestätigung_GF_anlegen	<GF_anlegen>FehlerNr + (Geschäftsfall) * optionaler Teil nur bei FehlerNr = 0 *
Bestätigung_GF_löschen	<GF_löschen>FehlerNr + ("Geschäftsfall wurde gelöscht.") * optionaler Teil nur bei FehlerNr = 0 *
GF-Bezeichnung	GF_Nr + (<GF_>Name)
Geschäftsfallinformationen	Geschäftsfall
Buchungsinformationen	<Buchungs>Datum + Beleg_Kommentar + ["direkt" + Kontenliste "über Geschäftsfall" + GF_Nr + Buchungsbeträge] * die Kardinalität von Buchungsbeträge entspricht der Kardinalität der im ausgewählten Geschäftsfall angelegten {Buchungssatzzeile_blanko} *
Buchungsauswahl	["direkt" "über Geschäftsfall"]
Bestätigung_Buchung	<Buchung>FehlerNr + ("Buchungssatz unter Belegnummer " + Beleg_ID + " erfolgreich gebucht.") * optionaler Teil nur bei FehlerNr = 0 *
AB-Änderungsdaten	AB-Änderungsliste + <B_>Name
AB-Änderungsliste	{AB-Änderungslisteneintrag}
AB-Änderungslisteneintrag	Kto_Nr + (<Kto_>Name) + <AB_>Betrag
Bestätigung_EBilanzerstellung	<EBilanzerstellung>FehlerNr + (Bilanz) * optionaler Teil nur bei FehlerNr = 0 *
SBilanz-Erstellungsauftrag	<B_>Name
Bestätigung_SBilanzerstellung	<SBilanzerstellung>FehlerNr + (Bilanz) * optionaler Teil nur bei FehlerNr = 0 *
Bilanz-Bezeichnung	<B_>Datum + (<B_>Name) + B_Art
Bilanzinformationen	Bilanz
Archiv-Löschungsauftrag	<A_>Datum
Bestätigung_Archivlöschung	<A_Daten_löschen>FehlerNr + ("Die Datenbestände wurden gelöscht.") * optionaler Teil nur bei FehlerNr = 0 *
FehlerNr	gZahl * 4stellig, repräsentiert den informierenden Text aus der Fehlercodetabelle *
benannte ERD-Elemente	
trägt_ein_in	@GF_Nr + <Buchungs>Datum + Beleg_Kommentar + Beleg_ID + Buchungsbeträge + <GB_>Datumsflag * <GB_>Datumsflag = 0 *
bucht_auf	@GF_Nr + Beleg_ID + Buchungsbeträge + 2{Kto_Nr} * Buchungsbeträge und {Kto_Nr} treten mit gleicher Kardinalität auf *
eröffnet	@<B_>Datum + B_Art + [@Kto_Nr <GB_>Datumsflag]
abgeschlossen_zu	@Kto_Nr + <B_>Datum + B_Art * B_Art = "SB" *
beendet	@<B_>Datum + B_Art + @<GB_>Datumsflag
Buchungsbeträge	2{<Buchungs>Betrag}

1.2.2.5 Beziehungen zwischen Speichern (ERD)



1.2.2.6 Prozessspezifikationen

1.2.2.6.1 Kto_Änderung

PSpec: **Kto_Änderung**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

```
begin
  if Kto-Änderungsauswahl = "Konto anlegen" then
    Rufe Prozess Kto_anlegen (Kto-Änderungsdaten)
    Setze Bestätigung_Kto-Änderung = zurückgegebener Wert
  else * Kto_Änderungsauswahl = "Konto löschen" *
    Rufe Prozess Kto_löschen (Kto-Änderungsdaten)
    Setze Bestätigung_Kto-Änderung = zurückgegebener Wert
  end if
  Gib Bestätigung_Kto-Änderung aus
end
```

1.2.2.6.2 Kto_löschen

PSpec: **Kto_löschen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: Wird gerufen von Oberprozess Kto_Änderung

```
begin
  Finde in Konten das Konto mit Konto.Kto_Nr = Kto_Nr * eingehender Datenfluss *
    and Konto.Kto_Datumsflag = 0
  if Konto nicht gefunden then
    Setze Bestätigung_Kto_löschen := "Konto nicht gefunden"
  else * Konto gefunden *
    * SBK und EK dürfen niemals gelöscht werden *
    if (Konto.Kto_Wertigkeit = 0 oder Konto.Kto_Art = "EK") then
      Setze Bestätigung_Kto_löschen := "Konto darf nicht gelöscht werden"
    * nur leere Konten dürfen gelöscht werden *
    else if (Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme = 0
      and Konto.Kto_Inhalt.Haben_Kto_Seite.Kto_SeitenSumme = 0) then
      Finde in Konten alle direkten und indirekten Unterkonten des Kontos
      Lösche alle gefundenen Unterkonten
      Lösche das Konto
      Setze Bestätigung_Kto_löschen := "Konto wurde mitsamt seiner Unterkonten gelöscht"
    else
      Setze Bestätigung_Kto_löschen := "Konto darf nicht gelöscht werden"
    end if
  end if
  Übergib Bestätigung_Kto_löschen an Prozess Kto_Änderung * ausgehender Datenfluss *
end
```


1.2.2.6.4 Konto_anzeigen

PSpec: **Konto_anzeigen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

begin

Finde in Konten das Konto mit Konto.Kto_Nr = Kto_Nr **and** Konto.Kto_Datumsflag = Kto_Datumsflag

if Konto gefunden **then**

Setze Kontoinformationen := Konto

else

Setze Kontoinformationen := "Konto nicht gefunden"

end if

Gib Kontoinformationen aus * ausgehender Datenfluss *

end

1.2.2.6.5 Grundbuch_anzeigen

PSpec: **Grundbuch_anzeigen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

begin

Finde in Grundbücher das Grundbuch mit Grundbuch.GB_Datumsflag = GB_Datumsflag

* eingehender Datenfluss *

if Grundbuch gefunden **then**

Setze Grundbuchinformationen := Grundbuch

else

Setze Grundbuchinformationen := "Grundbuch nicht gefunden"

end if

Gib Grundbuchinformationen aus * ausgehender Datenfluss *

end

1.2.2.6.6 GF_Änderung

PSpec: **GF_Änderung**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

begin

if GF-Änderungsauswahl = "Geschäftsfall anlegen" **then**

Rufe Prozess GF_anlegen(GF-Änderungsdaten)

Setze Bestätigung_GF-Änderung = zurückgegebener Wert

else * GF-Änderungsauswahl = "Geschäftsfall löschen" *

Rufe Prozess GF_löschen(GF-Änderungsdaten)

Setze Bestätigung_GF-Änderung = zurückgegebener Wert

end if

Gib Bestätigung_GF-Änderung aus

end

1.2.2.6.7 GF_anlegen

PSpec: **GF_anlegen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: Wird gerufen von Oberprozess GF_Änderung

begin

```
Überprüfe, ob in Geschäftsfälle ein Geschäftsfall existiert mit Geschäftsfall.GF_Nr = GF_Nr
                                                    * eingehender Datenfluss *
  or Geschäftsfall.GF_Name = GF_Name * eingehender Datenfluss *
if Geschäftsfall existiert then
  Setze Bestätigung_GF_anlegen :=
    "Geschäftsfall kann nicht angelegt werden, Geschäftsfallnummer/-name bereits vergeben"
else * Geschäftsfall existiert noch nicht und kann angelegt werden *
  Erstelle in Geschäftsfälle neuen Geschäftsfall mit { * eingehender Datenfluss *
    Geschäftsfall.GF_Nr := GF_Nr
    Geschäftsfall.GF_Name := GF_Name
    Geschäftsfall.GF_Vorgangsdaten := GF_Vorgangsdaten
  }
  Setze Bestätigung_GF_anlegen := "Geschäftsfall wurde angelegt"
end if
Übergib Bestätigung_GF_anlegen an Prozess GF_Änderung * ausgehender Datenfluss *
end
```

1.2.2.6.8 GF_löschen

PSpec: **GF_löschen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: Wird gerufen von Oberprozess GF_Änderung

begin

```
Finde in Geschäftsfälle den Geschäftsfall mit Geschäftsfall.GF_Nr = GF_Nr * eingehender Datenfluss *
if Geschäftsfall gefunden then
  Lösche Geschäftsfall
  Setze Bestätigung_GF_löschen := "Geschäftsfall wurde gelöscht"
else
  Setze Bestätigung_GF_löschen := "Geschäftsfall nicht gefunden"
end if
Übergib Bestätigung_GF_löschen an Prozess GF_Änderung * ausgehender Datenfluss *
end
```

1.2.2.6.9 Geschäftsfall_anzeigen

PSpec: **Geschäftsfall_anzeigen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

begin

Finde in Geschäftsfälle den Geschäftsfall mit Geschäftsfall.GF_Nr = GF_Nr * eingehender Datenfluss *

if Geschäftsfall gefunden **then**

 Setze Geschäftsfallinformationen := Geschäftsfall

else

 Setze Geschäftsfallinformationen := "Geschäftsfall nicht gefunden"

end if

Gib Geschäftsfallinformationen aus * ausgehender Datenfluss *

end

1.2.2.6.10 Buchung

PSpec: **Buchung**

Datum: 10.05.2003

Bearbeiter: D. Brückner

Voraussetzung: --

begin

if Buchungsinformationen.Buchungsauswahl = "über Geschäftsfall" **then**

Rufe Prozess GF_Daten_holen (Buchungsinformationen)

else

Rufe Prozess buchen_direkt (Buchungsinformationen)

end if

end

1.2.2.6.11 GF_Daten_holen

PSpec: GF_Daten_holen

Datum: 10.05.2003
Bearbeiter: D. Brückner

Voraussetzung: Wird durch Oberprozess Buchung gerufen

begin

Finde in Geschäftsfälle den Geschäftsfall mit Buchungsinformationen.GF_Nr

if Geschäftsfall nicht gefunden **then**

Bestätigung_Buchung.FehlerNr := "Geschäftsfall nicht gefunden"

Abbruch

else

* Auffüllen der Buchungsparameter mit Konteninformationen aus Vorlage *

* jedem übergebenen Betrag wird eine Blankozeile zugeordnet *

für jede Buchungssatzzeile_blanko in Geschäftsfall führe aus

begin verbundanweisung

Suche nächsten Betrag in Buchungsinformationen

if keinen Betrag gefunden **then**

Bestätigung_Buchung.FehlerNr := "Beträge stimmen nicht mit GF überein"

Abbruch

else

Füge ein in Buchungsinformationen Buchungssatzzeile mit

Buchungssatzzeile_blanko + Betrag

end if

end verbundanweisung

if weiterer Betrag vorhanden **then**

Bestätigung_Buchung.FehlerNr := "Beträge stimmen nicht mit GF überein"

Abbruch

else

Rufe Prozess buchen_direkt (Buchungsinformationen)

end if

end if

end

1.2.2.6.12 buchen_direkt

PSpec: **buchen_direkt**

Datum: 10.05.2003
Bearbeiter: D. Brückner

Voraussetzung: Wird gerufen durch Oberprozess Buchung oder Prozess GF_Daten_holen

begin

```
Erzeuge aus Datum und Zeit eine Beleg_ID
lege neuen Grundbucheintrag an mit {
  Grundbucheintrag.Datum := Buchungsinformationen.Datum
  Grundbucheintrag.Beleg_ID := Beleg_ID
  Grundbucheintrag.Beleg_Kommentar := Buchungsinformationen.Beleg_Kommentar
  Grundbucheintrag.Kontenliste := Buchungsinformationen.Kontenliste
}
* Vergleich der Kontoseiten mit zwei temp. Hilfsvariablen *
Berechne Abgänge := (Summe Beträge für alle Konten in Grundbucheintrag mit
  Buchungssatzzeile_blanko.betr_Kto_Seite = "Soll")
Berechne Zugänge := (Summe Beträge für alle Konten in Grundbucheintrag mit
  Buchungssatzzeile_blanko.betr_Kto_Seite = "Haben")
if Abgänge <> Zugänge then
  Bestätigung_Buchung.FehlerNr := "Buchungsfehlbetrag"
  Abbruch
end if
* nun folgt die Buchung auf den Konten *
für jeden Kontenlisteneintrag in Grundbucheintrag führe aus
begin verbundanweisung
  Suche in Konten Konto mit Kto_Nr = Kontenlisteneintrag.Buchungssatzzeile_blanko.Kto_Nr
  if Konto nicht gefunden then
    Bestätigung_Buchung.FehlerNr := "Konto nicht vorhanden"
    Abbruch
  else if Kontenlisteneintrag..Buchungssatzzeile_blanko.betr_Kto_Seite = "Soll" then
    Erstelle in Konto.Kto_Inhalt.Soll_Kto_Seite neuen Buchungseintrag mit Beleg_ID
    Aktualisiere Konto.Kto_Inhalt.Soll_Kto_Seite.Summe mit Kontenlisteneintrag.Betrag
  else
    Erstelle in Konto.Kto_Inhalt.Haben_Kto_Seite neuen Buchungseintrag mit Beleg_ID
    Aktualisiere Konto.Kto_Inhalt.Haben_Kto_Seite.Summe mit Kontenlisteneintrag.Betrag
  end if
end if
end verbundanweisung
finde Grundbuch mit Datumsflag = 0
füge ein Grundbucheintrag in gefundenes Grundbuch
Bestätigung_Buchung.FehlerNr := 0
Bestätigung_Buchung.Beleg_ID := Buchungseintrag.Beleg_ID
end
```

1.2.2.6.13 EBilanz_Erstellung

PSpec: **EBilanz_Erstellung**

Datum: 07.05.2003
Bearbeiter: D. Petersen

Voraussetzung: --

begin

* neue Eröffnungsbilanz erstellen *

Erstelle in Bilanzen neue EBilanz mit {

EBilanz.B_Datum = heute() * PHP Funktion *

EBilanz.B_Name = B_Name * eingehender Datenstrom *

EBilanz.B_Art = "EB"

EBilanz.B_Inhalt = ""

}

* aktiv und passiv Summen der Eingangsdaten bilden *

summe_aktiv = summe_passiv = 0

Solange es noch nicht betrachtete AB-Änderungslisteneintrag in AB-Änderungsdaten gibt

* eingehender Datenstrom *

do

Finde in Konten das Konto mit Kto_Nr AB-Änderungslisteneintrag.Kto_Nr

if Konto.Kto_Art = "aktiv" **then**

Setze summe_aktiv += AB-Änderungslisteneintrag.AB_Betrag

else if Konto.Kto_Art = "passiv" **then**

Setze summe_passiv += AB-Änderungslisteneintrag.AB_Betrag

end if

end if

end do

Setze saldo := summe_aktiv - summe_passiv

Setze EBilanz.B_Inhalt.Aktiva_B_Seite := summe_aktiv

Setze EBilanz.B_Inhalt.Passiva_B_Seite := summe_aktiv

Finde Konto in Konten mit Konto.Kto_Art = "EK"

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Nr := 0000

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Name := "AB"

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo

Setze EBilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Nr := Kto_Nr vom EK-Konto

Setze EBilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Name := Kto_Name vom EK-Konto

Setze EBilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Betrag := saldo

Sortiere AB-Änderungsdaten aufsteigend nach Kto_Nr

Solange es noch nicht betrachtete AB-Änderungslisteneintrag in AB-Änderungsdaten gibt

* eingehender Datenstrom *

do

Rufe Prozess AB_Änderung_vortragen (AB-Änderungslisteneintrag)

end do

Setze Bestätigung_EBilanzerstellung := "Bilanz wurde erstellt"

end

1.2.2.6.14 AB_Änderung_vortragen

PSpec: AB_Änderung_vortragen

Datum: 07.05.2003
Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Prozess EBilanz_Erstellung

begin

Finde in Konten das Konto mit der Kto_Nr = AB-Änderungslisteneintrag.Kto_Nr

if Konto.Kto_Art = "aktiv" **then**

Setze EBilanz.B_Inhalt.Aktiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Nr := AB-Änderungslisteneintrag.Kto_Nr
Setze EBilanz.B_Inhalt.Aktiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Name := AB-Änderungslisteneintrag.Kto_Name
Setze EBilanz.B_Inhalt.Aktiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Betrag := AB-Änderungslisteneintrag.AB_Betrag
Finde in Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseinträge den Buchungseintrag mit GegenKto_Nr = 0000
Setze Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegen_Betrag := AB-Änderungslisteneintrag.AB_Betrag
Setze Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme := AB-Änderungslisteneintrag.AB_Betrag

else if Konto.Kto_Art = "passiv" **then**

Setze EBilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Nr := AB-Änderungslisteneintrag.Kto_Nr
Setze EBilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Name := AB-Änderungslisteneintrag.Kto_Name
Setze EBilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.B_Posten.BP_Betrag := AB-Änderungslisteneintrag.AB_Betrag
Finde in Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseinträge den Buchungseintrag mit GegenKto_Nr = 0000
Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegen_Betrag := AB-Änderungslisteneintrag.AB_Betrag
Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Kto_SeitenSumme := AB-Änderungslisteneintrag.AB_Betrag

end if

end if

end

1.2.2.6.15 Konten_suchen

PSpec: Konten_suchen

Datum: 07.05.2003
Bearbeiter: D. Petersen

Voraussetzung: --

begin

Finde in Bilanzen ein SBK mit Abschlussdatum von heute() *PHP Funktion*

if gefunden

Setze Bestätigung_SBilanzerstellung := "Bilanz heute bereits abgeschlossen"

else

* alle Konten ausser dem SBK abschließen *

Solange es noch ein Konto in Konten gibt mit Konto.Kto_Datumsflag = 0 **und** Konto.Kto_Wertigkeit != 0

do

Finde in Konten Konto mit der höchsten Wertigkeit, welches noch nicht abgeschlossen ist

Rufe Prozess Konto_abschließen_und_neuerstellen (Konto)

end do

Solange es noch ein Konto in Konten gibt mit Konto.Kto_Datumsflag = -1

do

Setze Konto.Kto_Datumsflag := 0

end do

Rufe Prozess Bilanz_erstellen(SBilanz-Erstellungsauftrag.B_Name) * eingehender Datenfluss *

Setze Bestätigung_SBilanzerstellung := "Bilanz wurde erstellt"

Rufe Prozess Grundbuch_abschließen_und_neuerstellen ()

endif

end

1.2.2.6.16 Konto_abschließen_und_neuerstellen

PSpec: **Konto_abschließen_und_neuerstellen**

Datum: 07.05.2003

Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Oberprozess Konten_suchen

begin

Setze soll := Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme

Setze haben := Konto.Kto_Inhalt.Haben_Kto_Seite.Kto_SeitenSumme

* vergleiche beide Summenseiten *

Setze saldo := absolut(soll - haben)

Finde in Konten Oberkonto von Konto

Finde in Grundbücher Grundbuch mit GB_Datumsflag = 0

* buchen *

if soll != haben **then**

if soll > haben **then**

Setze seite := "Haben"

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Kto_SeitenSumme := soll

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegenkonto_Nr := Konto.OberKto_Nr

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegenkonto_Name :=

Name von Konto.OberKto_Nr

Setze Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo

Setze Oberkonto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegenkonto_Nr := Konto.Kto_Nr

Setze Oberkonto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegenkonto_Name := Konto.Kto_Name

Setze Oberkonto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo

Setze Oberkonto.Kto_Inhalt.Soll_Kto_Seite.Kto_Seiten_Summe += saldo

else

Setze seite := "Soll"

Setze Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme := haben

Setze Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegenkonto_Nr := Konto.OberKto_Nr

Setze Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegenkonto_Name := Name von Konto.OberKto_Nr

Setze Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo

Setze Oberkonto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegenkonto_Nr := Konto.Kto_Nr

Setze Oberkonto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegenkonto_Name := Konto.Kto_Name

Setze Oberkonto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo

Setze Oberkonto.Kto_Inhalt.Haben_Kto_Seite.Kto_Seiten_Summe += saldo

endif

endif

* Grundbucheintrag *

Setze Grundbuch.GB_Inhalt.BelegID := aktuelles Datum + aktuelle Zeit

Setze Grundbuch.GB_Inhalt.Kommentar := "Abschlussbuchung"

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[1].Buchungssatzzeile_blanko.Kto_Nr :=
Konto.Kto_Nr

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[1].Buchungssatzzeile_blanko.Kto_Name :=
Konto.Kto_Name

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[1].Buchungssatzzeile_blanko.betr_Kto_Seite :=
seite

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[1].Buchungsbetrag := saldo

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[2].Buchungssatzzeile_blanko.Kto_Nr :=
Konto.OberKto.Kto_Nr

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[2].Buchungssatzzeile_blanko.Kto_Name :=
Name von Konto.OberKto.Kto_Nr

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[2].Buchungssatzzeile_blanko.betr_Kto_Seite :=
andere(seite)

Setze Grundbuch.GB_Inhalt.Kontenliste.Kontenlisteneintrag[2].Buchungsbetrag := saldo

Setze Konto.Kto_Datumsflag := heute() *PHP Funktion*

* neues leeres Konto erstellen *

Erstelle neues_Konto und setze neues_Konto := Konto

Setze neues_Konto.Kto_Inhalt := ""

Setze neues_Konto.Kto_Datumsflag := -1

```

* AB vortragen, wenn nötig *
if neues_Konto.Kto_Wertigkeit = 1 then
  if neues_Konto.Kto_Art = "aktiv" then *AB vortragen*
    Setze neues_Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.GegenKto_Nr := 0000
    Setze neues_Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.GegenKto_Name := "AB"
    Setze neues_Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo
    Setze neues_Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme := saldo
    if neues_Konto.Kto_Art = ("passiv" oder "EK") then *AB vortragen*
      Setze neues_Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Nr := 0000
      Setze neues_Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Name := "AB"
      Setze neues_Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.Gegen_Betrag := saldo
      Setze neues_Konto.Kto_Inhalt.Haben_Kto_Seite.Kto_SeitenSumme := saldo
    endif
  endif
endif
end

```

1.2.2.6.17 Grundbuch_abschließen_und_neuerstellen

PSpec: **Grundbuch_abschließen_und_neuerstellen**

Datum: 07.05.2003

Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Oberprozess Konten_suchen

```

begin
  * Grundbuch abschließen und neu erstellen *
  Finde in Grundbücher das Grundbuch mit Grundbuch.GB_Datumsflag = 0
  Setze Grundbuch.GB_Datumsflag = heute()
  Erstelle in Grundbücher neues Grundbuch mit {
    Grundbuch.GB_Datumsflag = 0
    Grundbuch.GB_Inhalt = ""
  }
end

```

1.2.2.6.18 Bilanz_erstellen

PSpec: Bilanz_erstellen

Datum: 07.05.2003
Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Oberprozess Konten_suchen

begin

* neue Bilanz erstellen *

Erstelle in Bilanzen neue Bilanz mit {

 Bilanz.B_Datum = heute() *PHP Funktion*

 Bilanz.B_Name = SBilanz-Erstellungsauftrag.B_Name *eingehender Datenfluss*

 Bilanz.B_Art = "SB"

 Bilanz.B_Inhalt = ""

}

* übertrage SBK-Inhalt in das Bilanz-Konto *

Suche in Konten das Konto mit Konto.Kto_Wertigkeit = 0 **and** Konto.Kto_Datumsflag = 0

Solange es noch nicht betrachtete Buchungseinträge in Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseinträge gibt **do**

 Suche Buchungseintrag mit Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.GegenKto_Nr mit der kleinsten GegenKto_Nr

 Setze Bilanz.B_Inhalt.Aktiva_B_Seite.B_Seiteninhalt.BP_Nr :=

 Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.GegenKto_Nr

 Setze Bilanz.B_Inhalt.Aktiva_B_Seite.B_Seiteninhalt.BP_Name :=

 Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.GegenKto_Name

 Setze Bilanz.B_Inhalt.Aktiva_B_Seite.B_Seiteninhalt.BP_Betrag :=

 Konto.Kto_Inhalt.Soll_Kto_Seite.Buchungseintrag.GegenKto_Betrag

end do

Setze Bilanz.B_Inhalt.Aktiva_B_Seite.B_SeitenSumme := Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme

Solange es noch nicht betrachtete Buchungseinträge in Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseinträge gibt **do**

 Suche Buchungseintrag mit Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Nr mit der kleinsten GegenKto_Nr

 Setze Bilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.BP_Nr :=

 Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Nr

 Setze Bilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.BP_Name :=

 Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Name

 Setze Bilanz.B_Inhalt.Passiva_B_Seite.B_Seiteninhalt.BP_Betrag :=

 Konto.Kto_Inhalt.Haben_Kto_Seite.Buchungseintrag.GegenKto_Betrag

end do

Setze Bilanz.B_Inhalt.Aktiva_B_Seite.B_SeitenSumme := Konto.Kto_Inhalt.Soll_Kto_Seite.Kto_SeitenSumme

Setze Konto.Kto_Datumsflag = heute() *PHP Funktion*

* neues leeres SBK erstellen *

Finde in Konten altes_Konto mit Kto_Art = "SBK"

Erstelle in Konten neues Konto mit {

 Konto.Kto_Nr = altes_Konto.Kto_Nr

 Konto.Kto_Name = altes_Konto.Kto_Name

 Konto.Kto_Art = "SBK"

 Konto.Kto_Datumsflag = 0

 Konto.Kto_Wertigkeit = 0

 Konto.Kto_OberKto_Nr = 0000

 Konto.Kto_Inhalt = ""

}

end

1.2.2.6.19 Bilanz_anzeigen

PSpec: **Bilanz_anzeigen**

Datum: 03.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

begin

Finde in Bilanzen die Bilanz mit Bilanz.B_Datum = B_Datum **and** Bilanz.B_Art = B_Art
* eingehender Datenfluss *

if Bilanz gefunden **then**

Setze Bilanzinformationen := Bilanz

else

Setze Bilanzinformationen := "Bilanz nicht gefunden"

end if

Gib Bilanzinformationen aus * ausgehender Datenfluss *

end

1.2.2.6.20 Archivdaten_Löschung

PSpec: **Archivdaten_Löschung**

Datum: 10.05.2003

Bearbeiter: A. Keusch

Voraussetzung: --

begin

Rufe Prozess A_GB_löschen()

Setze Bestätigung_Archivlöschung = zurückgegebener Wert

if (Bestätigung_Archivlöschung == 0) **then**

Rufe Prozess A_Kto_löschen()

Setze Bestätigung_Archivlöschung = zurückgegebener Wert

if (Bestätigung_Archivlöschung == 0) **then**

Rufe Prozess A_B_löschen()

Setze Bestätigung_Archivlöschung = zurückgegebener Wert

end if

end if

Gib Bestätigung_Archivlöschung aus

end

1.2.2.6.21 A_Kto_löschen

PSpec: A_Kto_löschen

Datum: 10.05.2003
Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Oberprozess Archivdaten_Löschung

begin

Finde in Konten ein Konto mit Konto.Kto_Datumsflag <= heute - 10 Jahre

if gefunden **then**

solange es in Konten noch ein Konto gibt mit Konto.Kto_Datumsflag <= heute() - 10 Jahre

do

Finde in Konten ein Konto mit Konto.Kto_Datumsflag <= heute - 10 Jahre

lösche Konto aus Konten

end do

Setze Bestätigung_A_Kto_löschen := "Erfolgreich gelöscht"

else

Setze Bestätigung_A_Kto_löschen := "Keine alten Konten gefunden"

end if

Übergib Bestätigung_A_Kto_löschen an Prozess Archivdaten_Löschung

end

1.2.2.6.22 A_GB_löschen

PSpec: A_GB_löschen

Datum: 10.05.2003
Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Oberprozess Archivdaten_Löschung

begin

Finde in Grundbücher ein Grundbuch mit Grundbuch.GB_Datumsflag <= heute - 10 Jahre

if gefunden **then**

solange es in Grundbücher noch ein Grundbuch gibt mit Grundbuch.GB_Datumsflag <= heute() - 10 Jahre

do

Finde in Grundbücher ein Grundbuch mit Grundbuch.GB_Datumsflag <= heute - 10 Jahre

lösche Grundbuch aus Grundbücher

end do

Setze Bestätigung_A_GB_löschen := "Erfolgreich gelöscht"

else

Setze Bestätigung_A_GB_löschen := "Keine alten Grundbücher gefunden"

end if

Übergib Bestätigung_A_GB_löschen an Prozess Archivdaten_Löschung

end

1.2.2.6.23 A_B_löschen

PSpec: A_B_löschen

Datum: 10.05.2003
Bearbeiter: D. Petersen

Voraussetzung: Wird gerufen von Oberprozess Archivdaten_Löschung

begin

Finde in Bilanzen eine Bilanz mit Bilanz.B_Datum <= heute - 10 Jahre

if gefunden **then**

solange es in Bilanzen noch eine Bilanz gibt mit Bilanz.B_Datum <= heute() - 10 Jahre

do

Finde in Bilanzen eine Bilanz mit Bilanz.B_Datum <= heute - 10 Jahre

lösche Bilanz aus Bilanzen

end do

Setze Bestätigung_A_B_löschen := "Erfolgreich gelöscht"

else

Setze Bestätigung_A_B_löschen := "Keine alten Bilanzen gefunden"

end if

Übergib Bestätigung_A_B_löschen an Prozess Archivdaten_Löschung

end

1.3 Definition der Nutzerschnittstelle

Generell:

Da die Nutzerschnittstelle in HTML programmiert wird, erfolgt die Darstellung über den Browser des Benutzers. Somit kann die Fenstergröße und die jeweilige Lage der Fenster vom Benutzer, mit Hilfe des Browsers, selbst eingestellt werden. Außerdem hat der Nutzer dadurch auch Einfluss auf die Schriftgröße bzw. Schriftart.

Grundsätzlich ist die Navigationsleiste auf der linken Seite ständig sichtbar. Nur der Inhalt auf der rechten Seite wechselt. Ausnahmen bilden diejenigen Seiten, bei denen es nicht möglich sein darf etwas anderes zu unternehmen, solange der Rechner mit einem Vorgang beschäftigt ist. Bei diesen Seiten wird keine Eingabe durch den Nutzer möglich sein, diesbezüglich sind die Button der Navigationsleiste inaktiv.

Die jeweiligen Seiten werden also immer im Grundfenster eingeblendet, einzige Ausnahme ist hier die Hilfeseite. Diese wird in einem sich neu öffnendem Fenster mit eigener Navigation zu sehen sein.

Farbgestaltung:

Um nicht von der eigentlichen Funktionalität des Systems abzulenken, wird auf besonders farbenfrohes Design verzichtet. Da ein deutlicher Kontrast zwischen Hintergrund und Schrift für das längere Arbeiten dienlich ist, wird der Hintergrund in 100% weiß und die Schrift in 100% schwarz sein. Lediglich der Text „Fehler“ und die jeweilige FehlerNr wird in 100% rot dargestellt, sofern die FehlerNr von Null verschieden ist.

1.3.1 Ein- und Ausgabegeräte


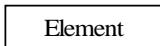




Als Eingabegeräte sind für das System die Tastatur und die Maus vorgesehen.

Ausgaben des Systems erfolgen grundsätzlich per Browser über den Monitor (color, 256 Farben min.).

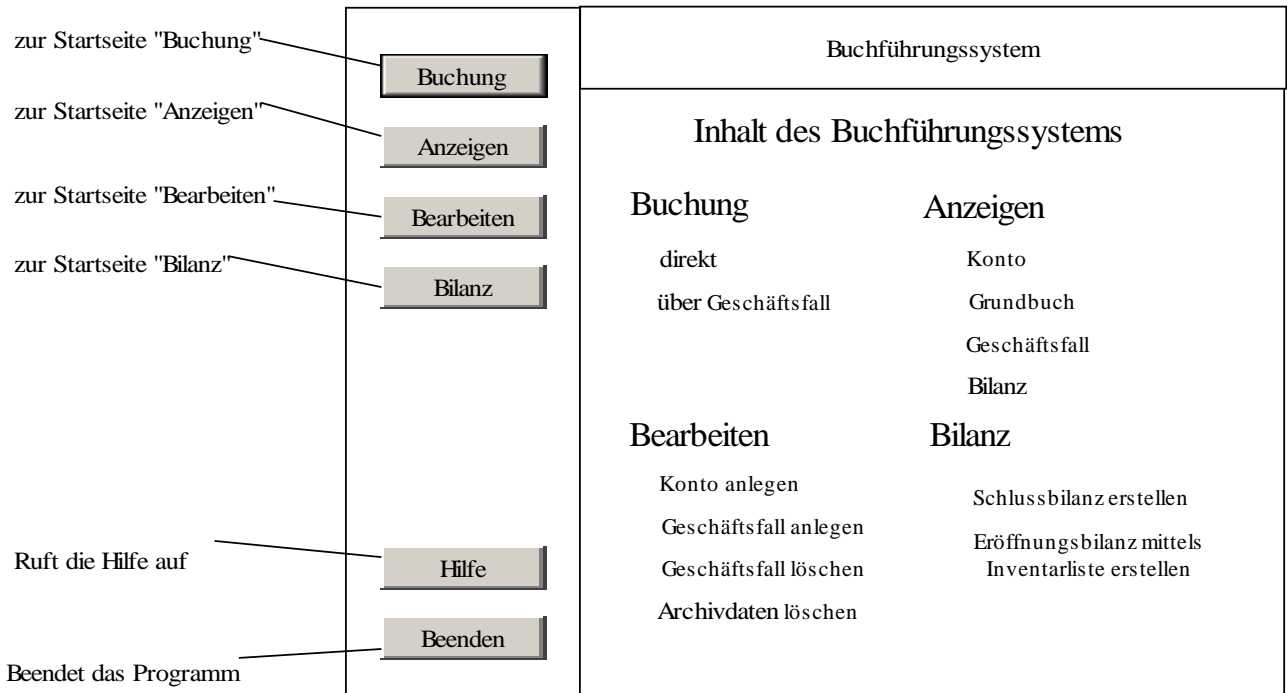
Eine spezielle Druckeinrichtung ist nicht vorgesehen, jedoch kann mit Hilfe des Browsers jederzeit ein Ausdruck der Seite erfolgen.

1.3.2 Layoutentwurf

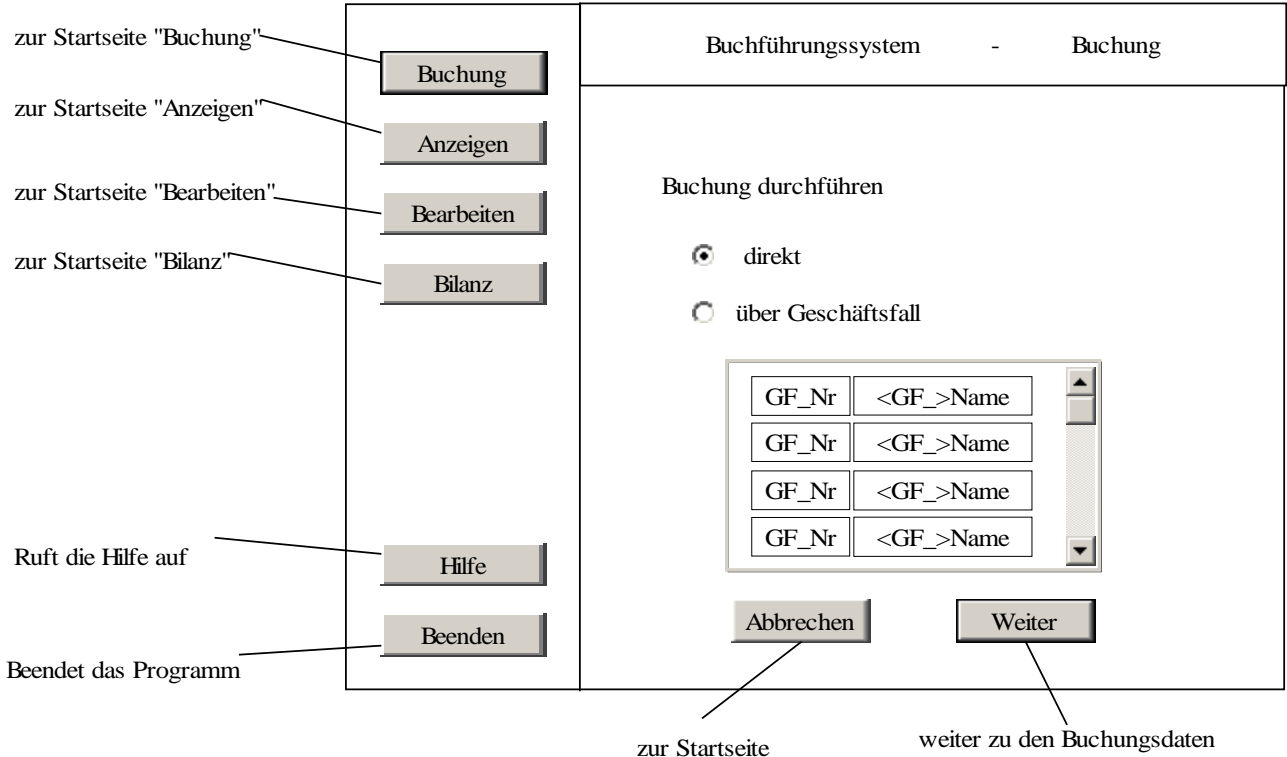
1.3.2.1 Legende zur Layoutdarstellung

	- Bedienknopf mit Beschriftung (mit gesondert angegebener Semantik)
	- Ausgabebereich und zugehöriges Element des Datenkataloges
	- Ein- und Ausgabebereich mit Datenkatalog-Element
	- Auswahlfelder; Nutzer kann zwischen verschiedenen Alternativen wählen
	- Auswahlbox; Nutzer kann zwischen verschiedenen Alternativen wählen
	- aufklappende Auswahlbox; Nutzer kann zwischen verschiedenen Alternativen wählen

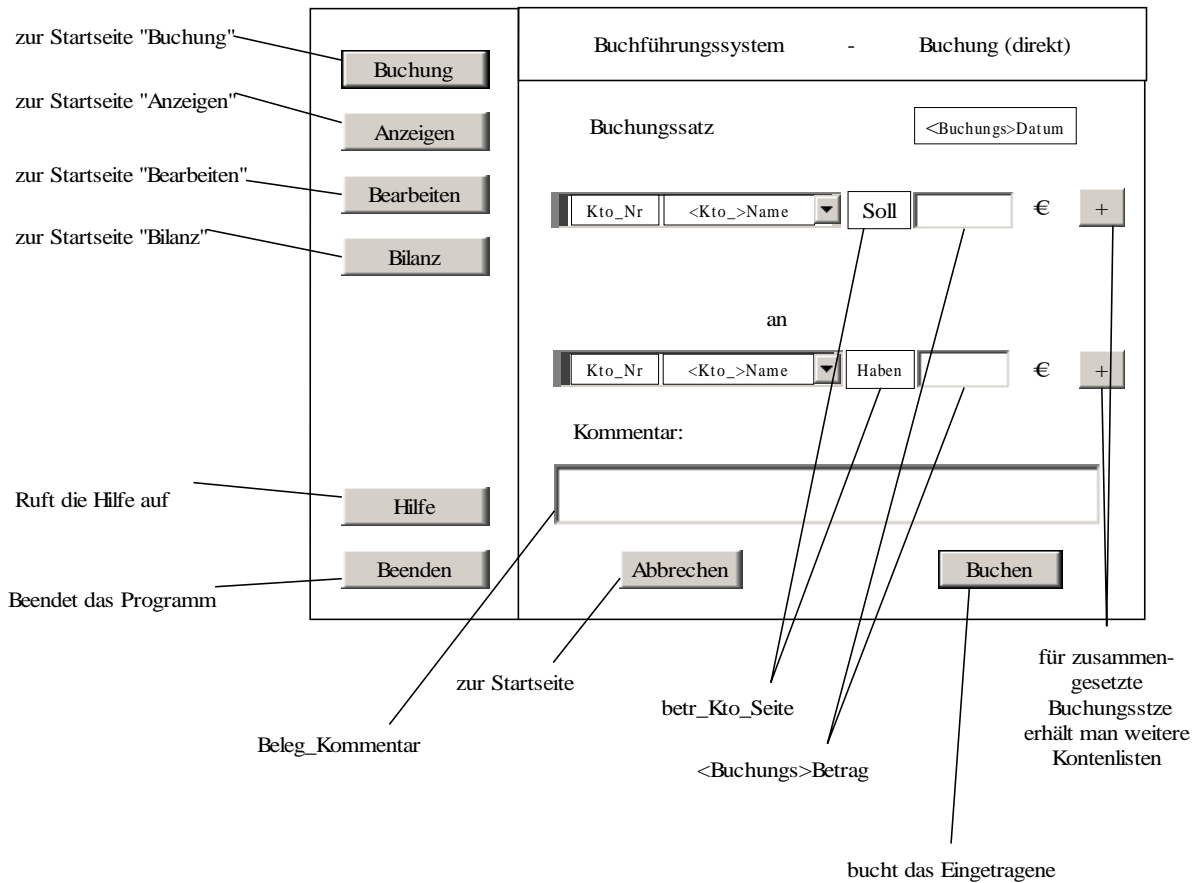
1.3.2.2 Startfenster



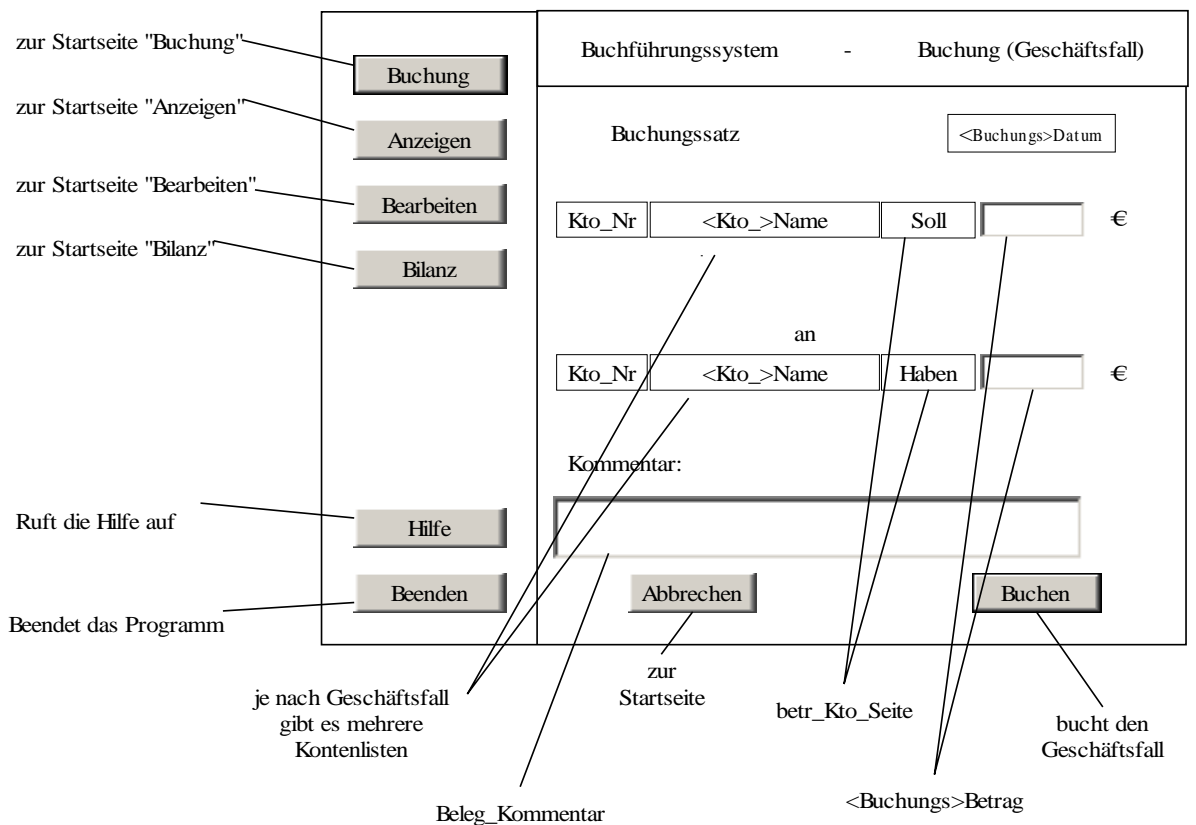
1.3.2.3 Fenster Buchung



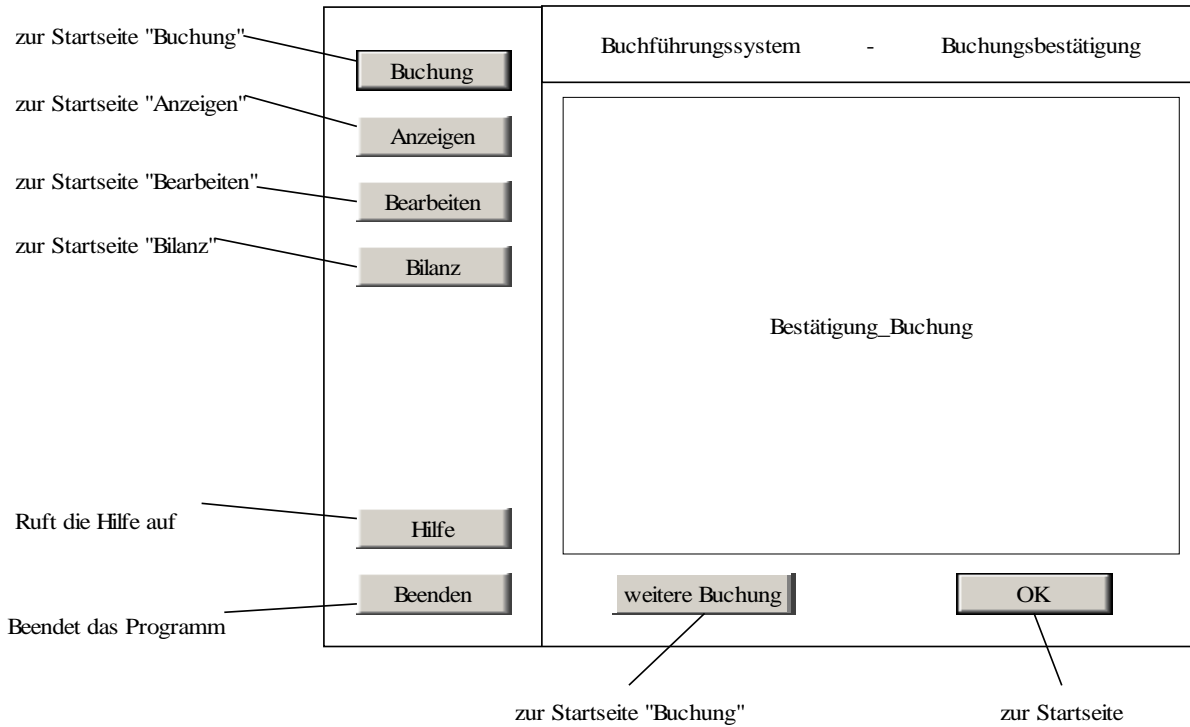
1.3.2.4 Fenster Buchung_direkt



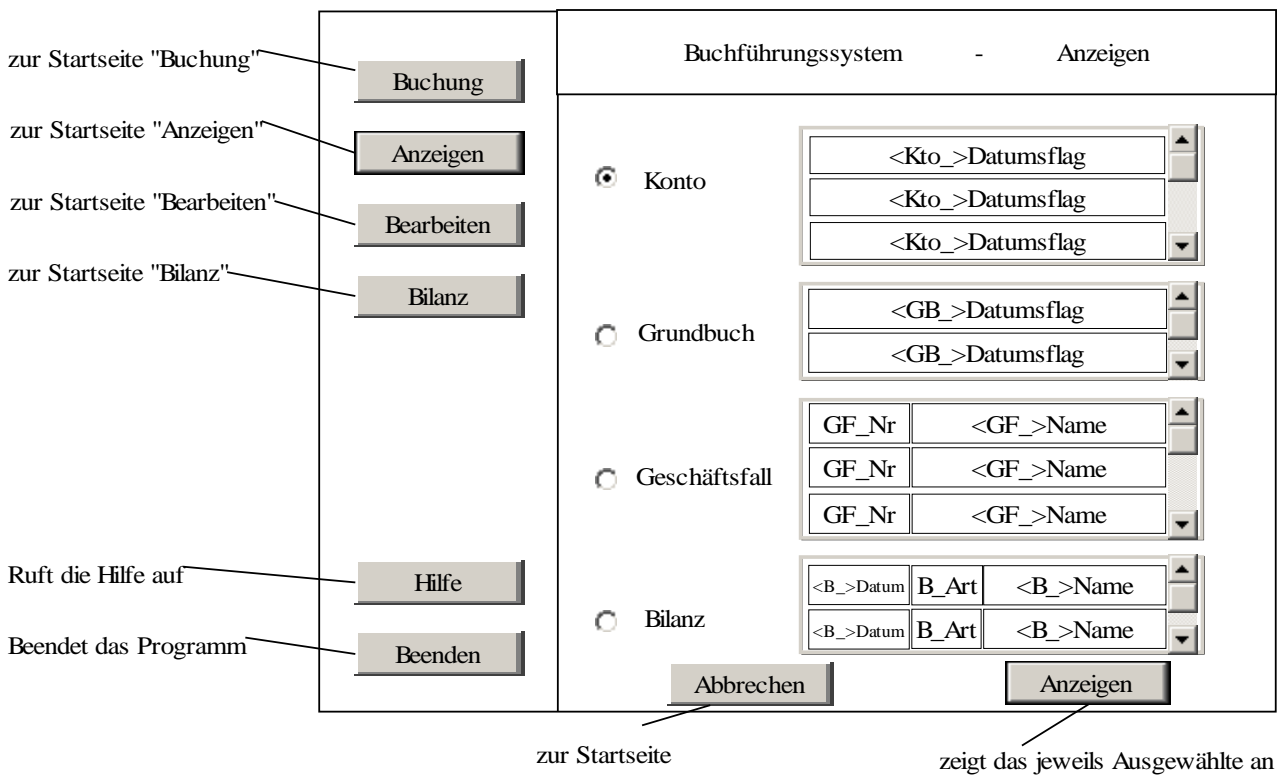
1.3.2.5 Fenster Buchung_über_Geschäftsfall



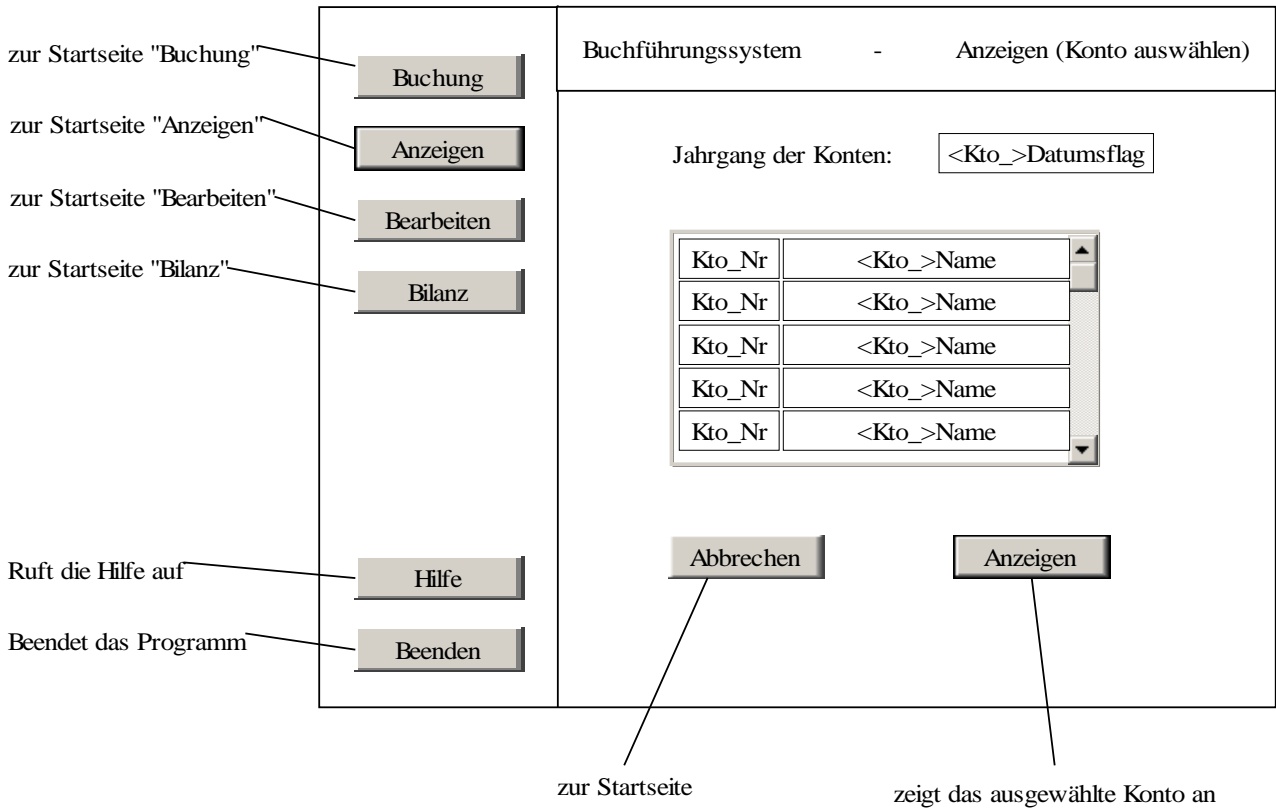
1.3.2.6 Fenster Bestätigung_Buchung



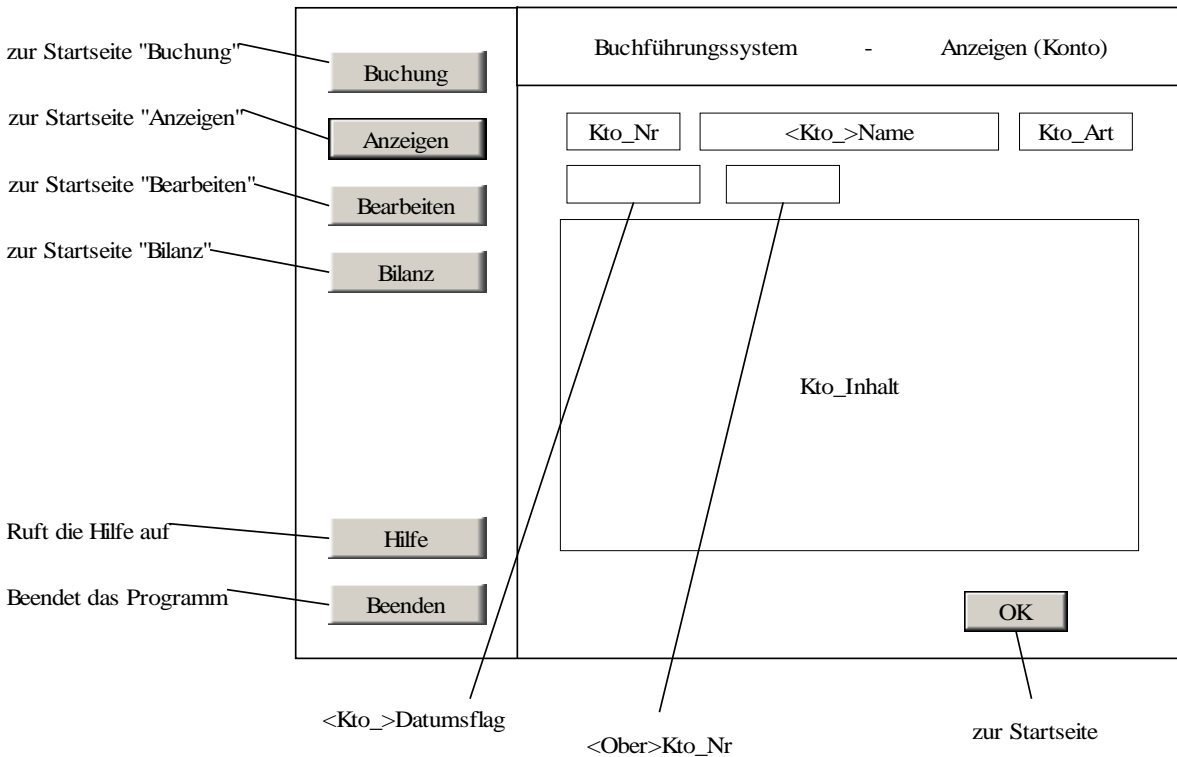
1.3.2.7 Fenster Anzeigen



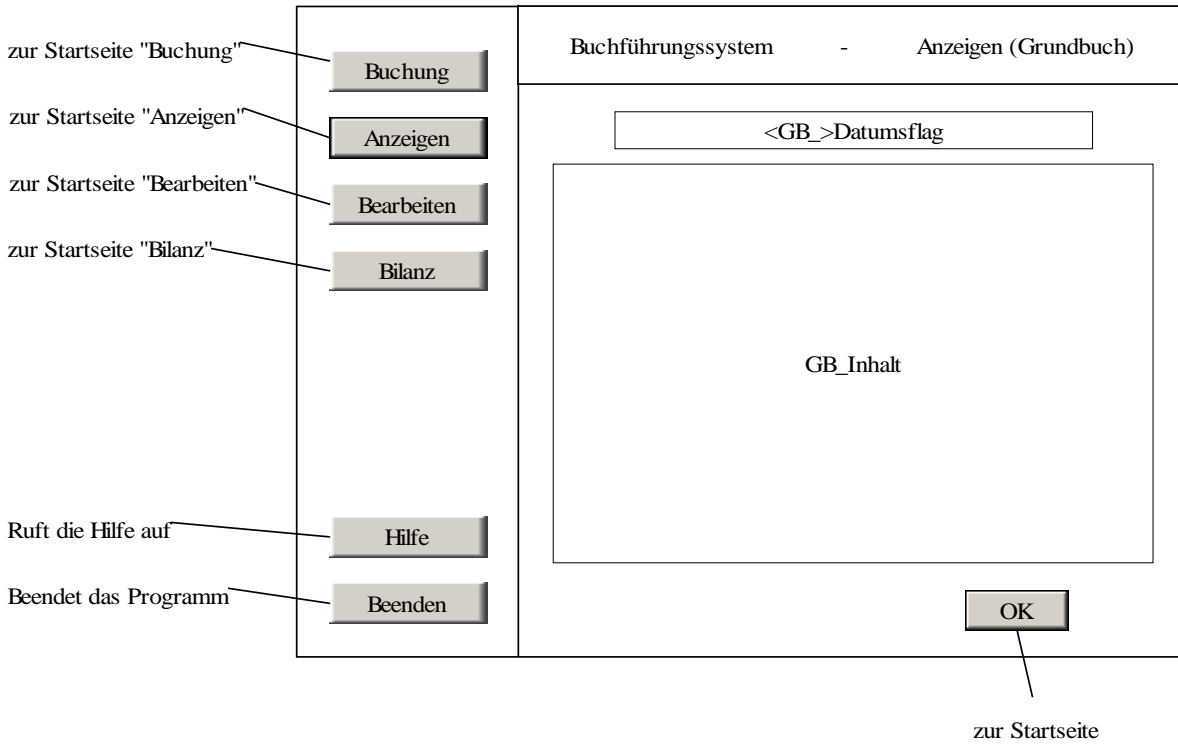
1.3.2.8 Fenster Konto_auswählen



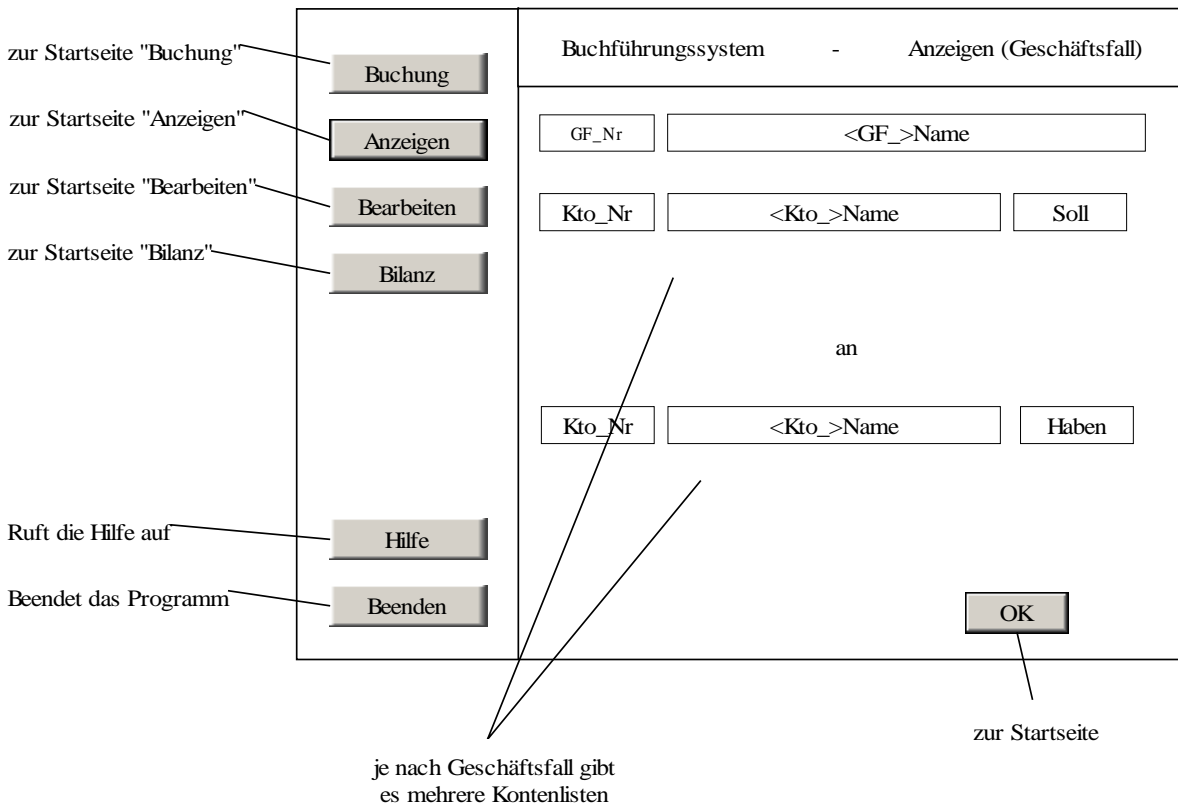
1.3.2.9 Fenster Konto_anzeigen



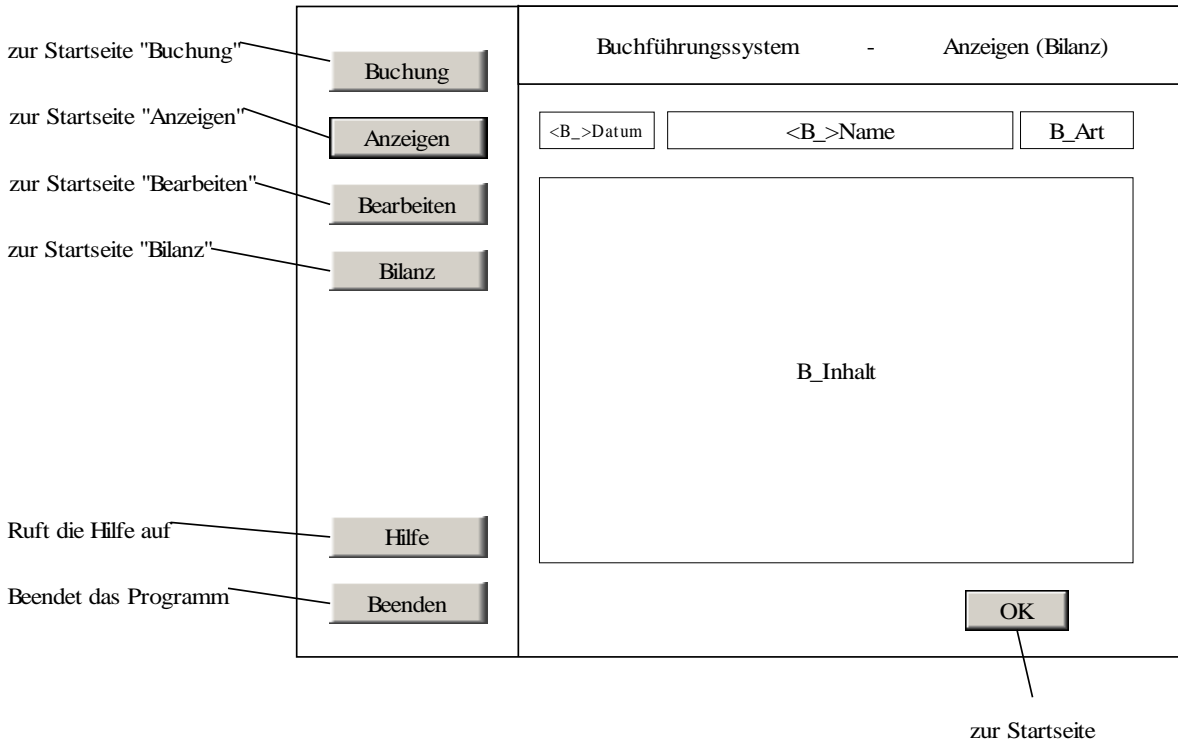
1.3.2.10 Fenster Grundbuch_anzeigen



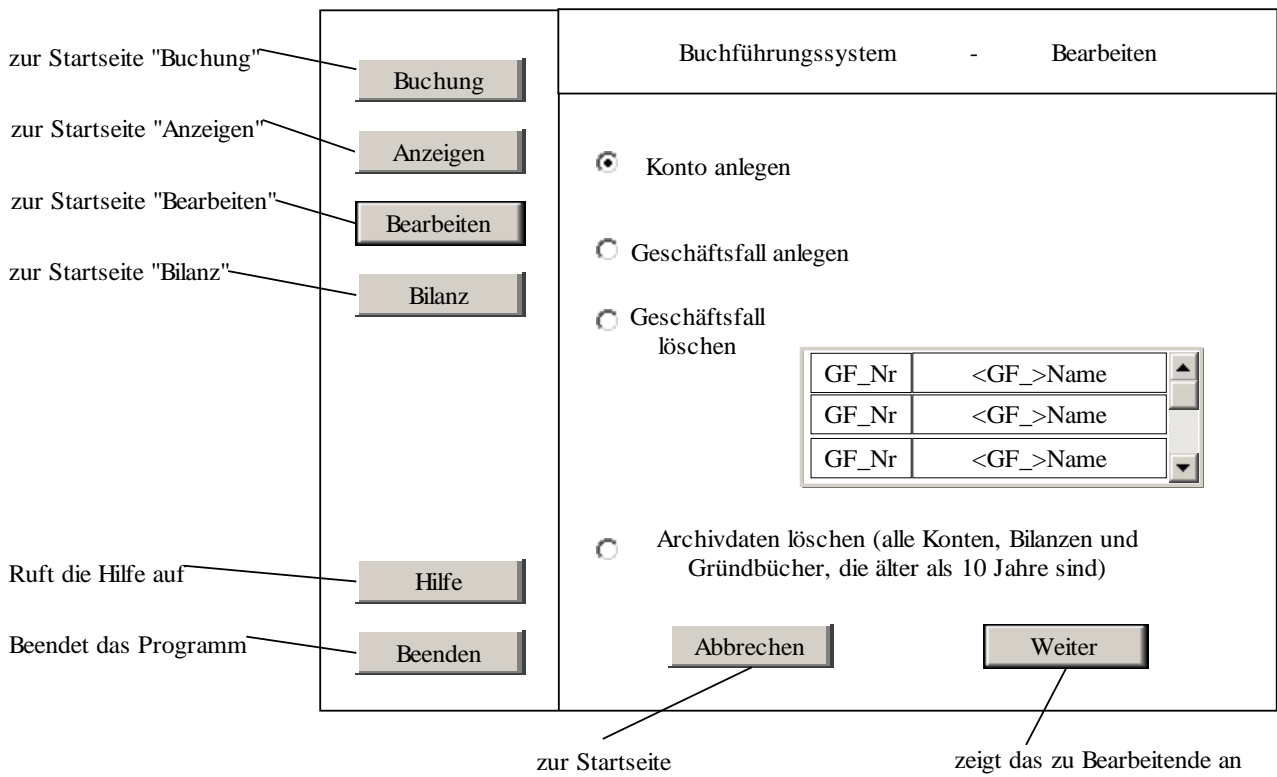
1.3.2.11 Fenster Geschäftsfall_anzeigen



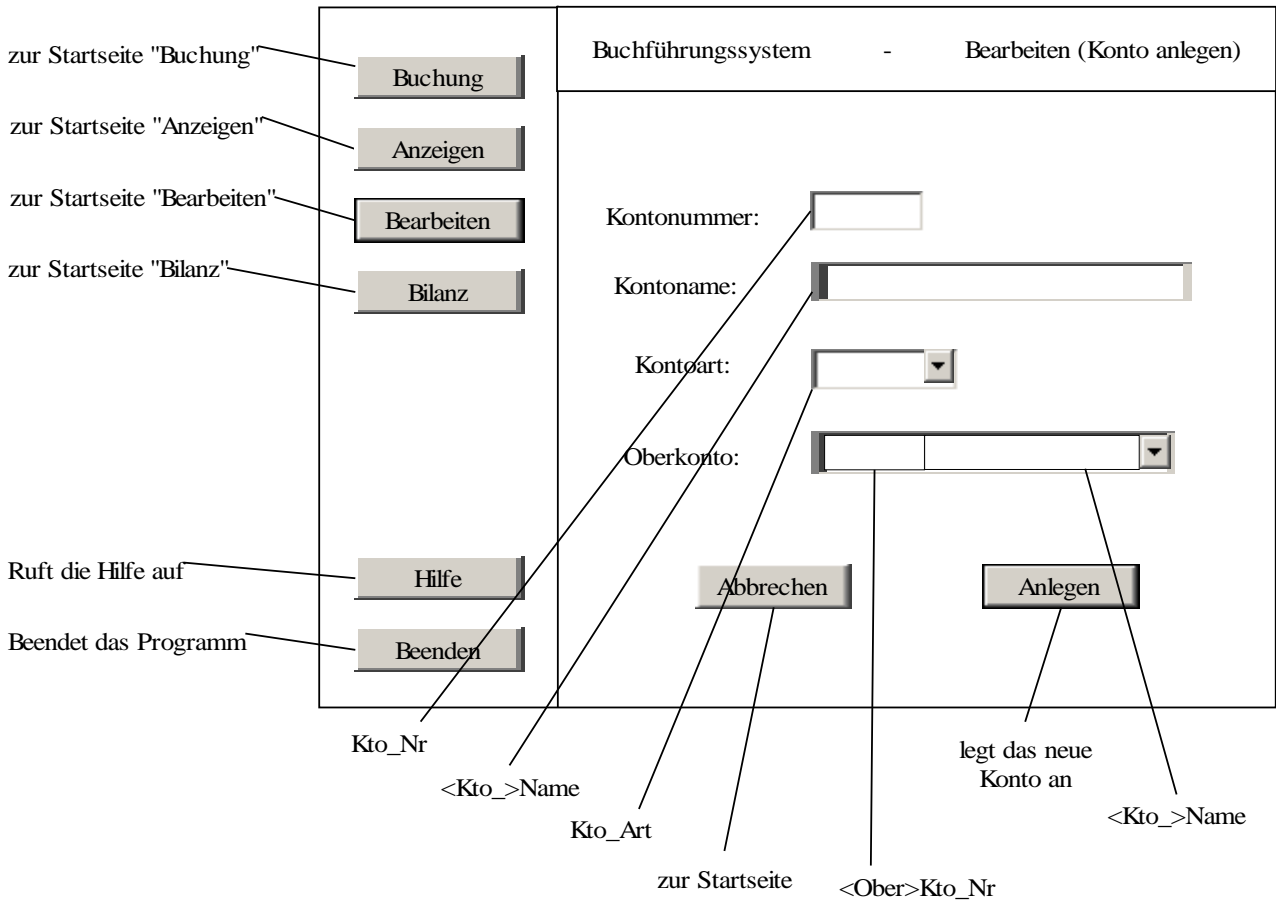
1.3.2.12 Fenster Bilanz_anzeigen



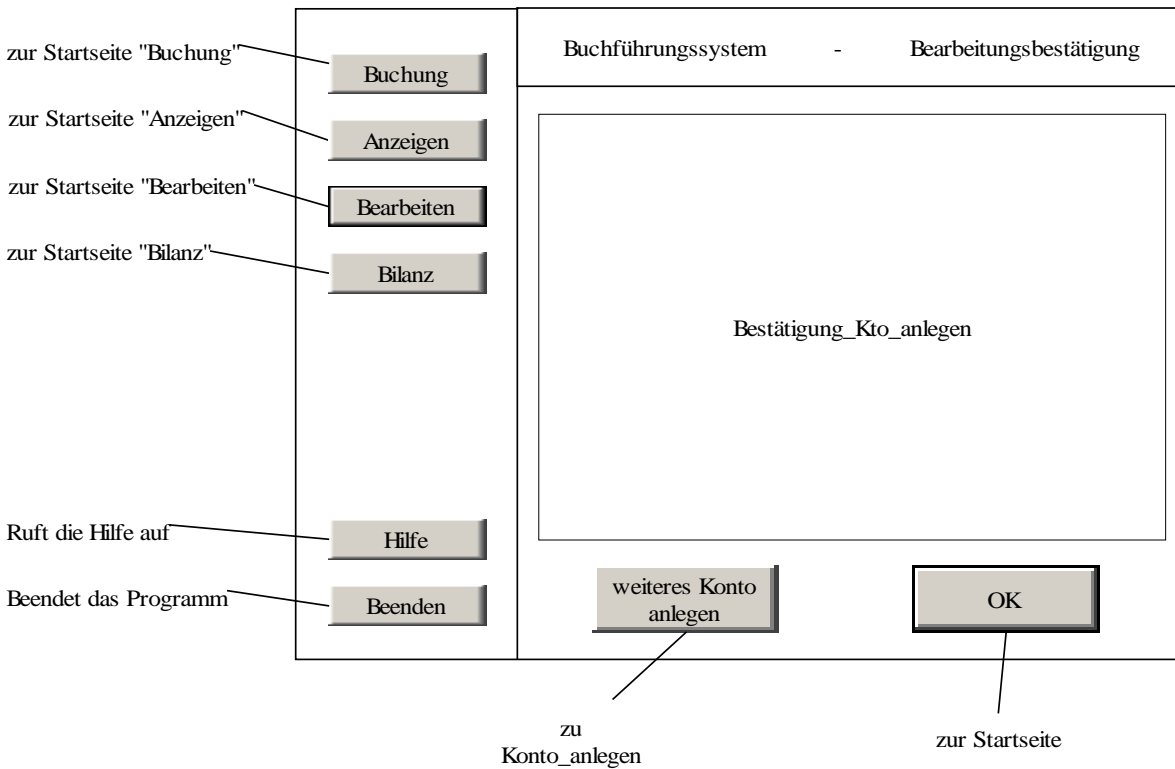
1.3.2.13 Fenster Bearbeiten



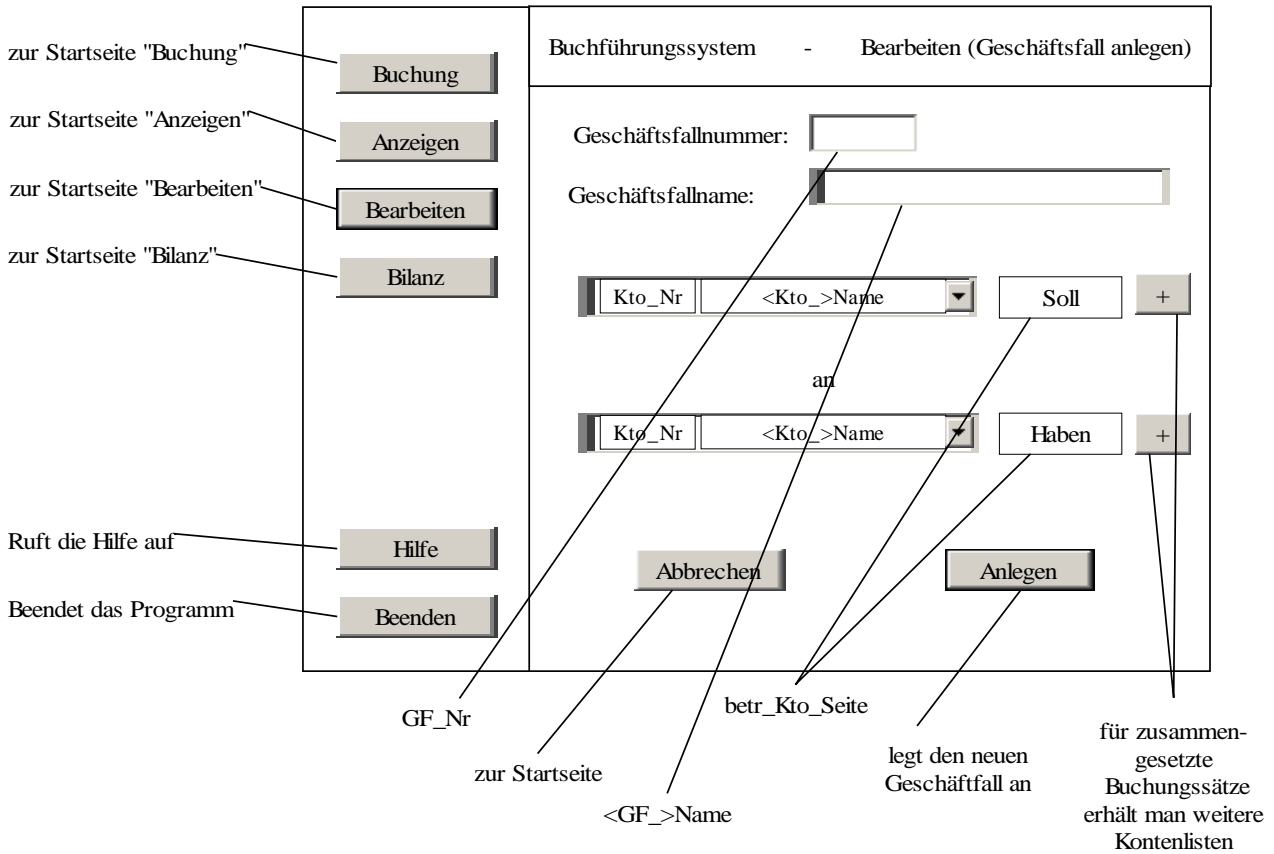
1.3.2.14 Fenster Konto_anlegen



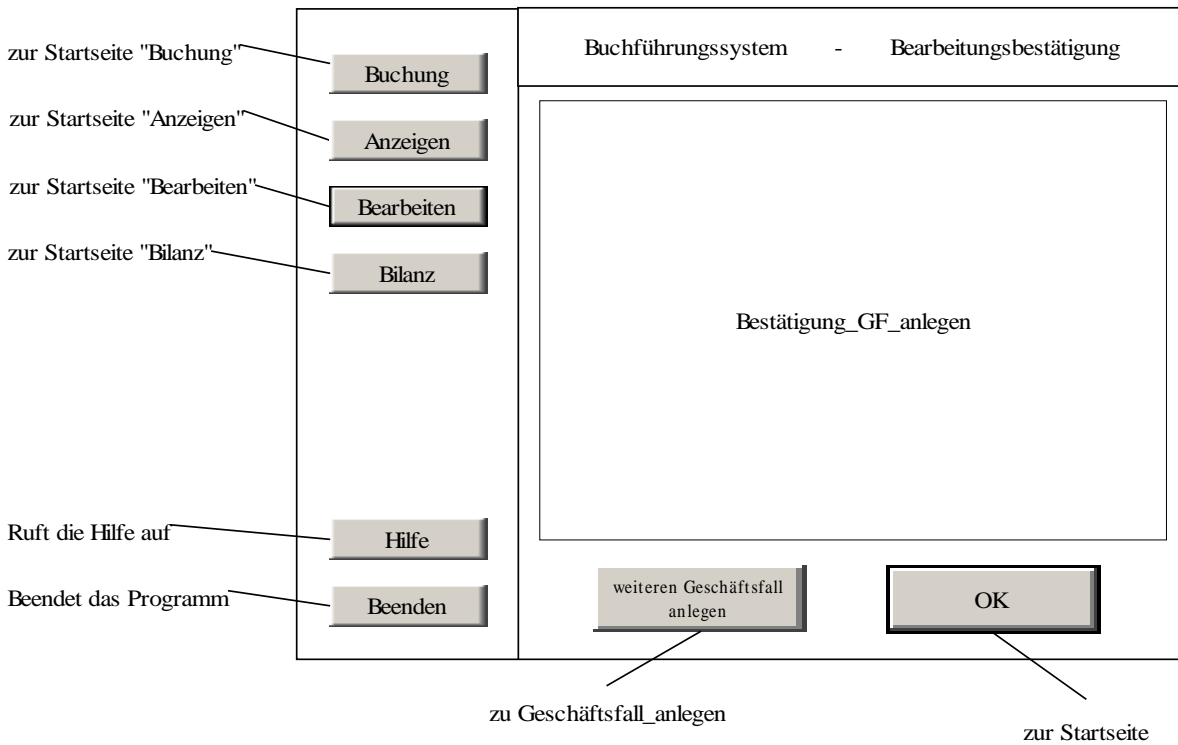
1.3.2.15 Fenster Bestätigung_Konto_anlegen



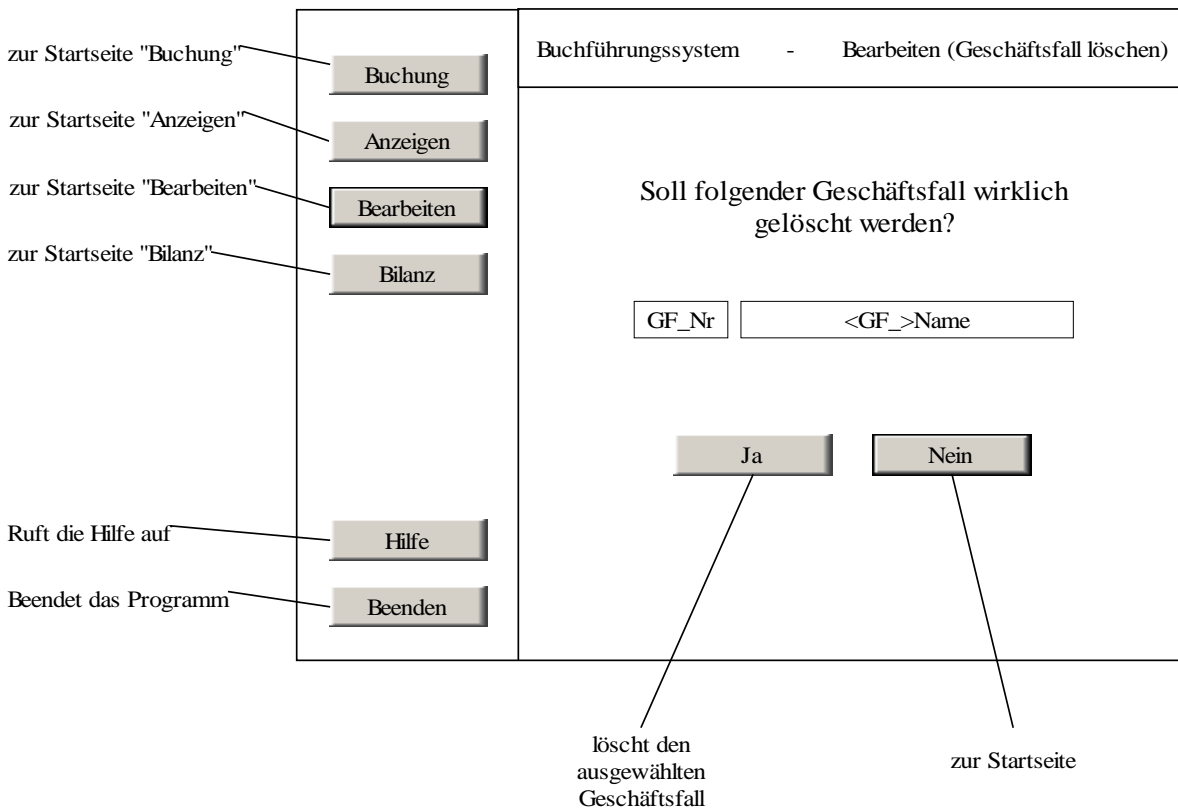
1.3.2.16 Fenster Geschäftsfall_anlegen



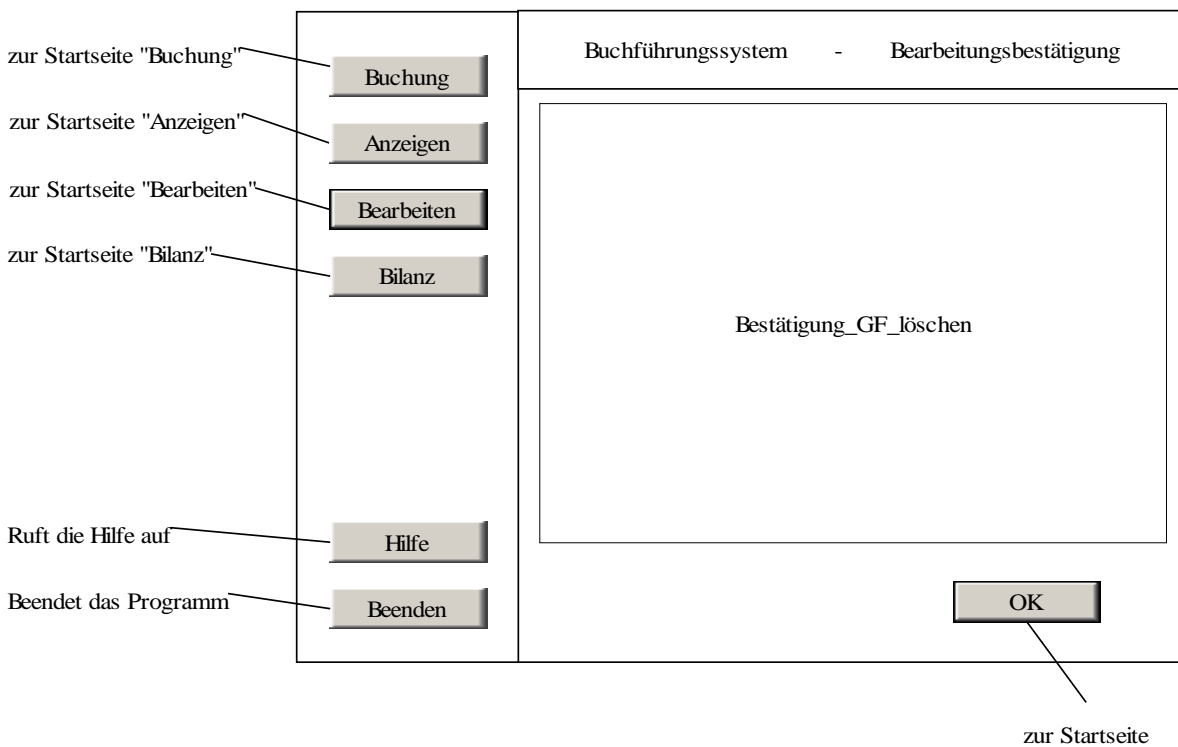
1.3.2.17 Fenster Bestätigung_Geschäftsfall_anlegen



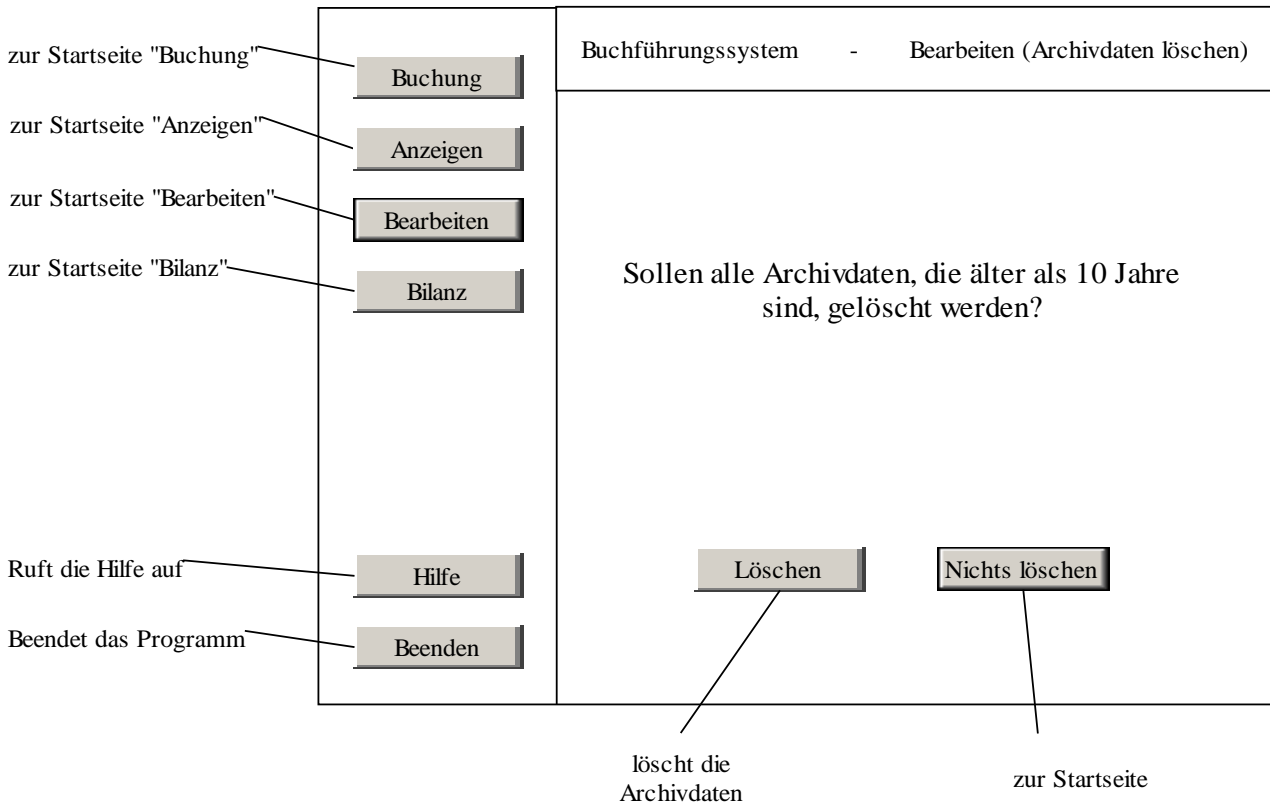
1.3.2.18 Fenster Geschäftsfall_löschen



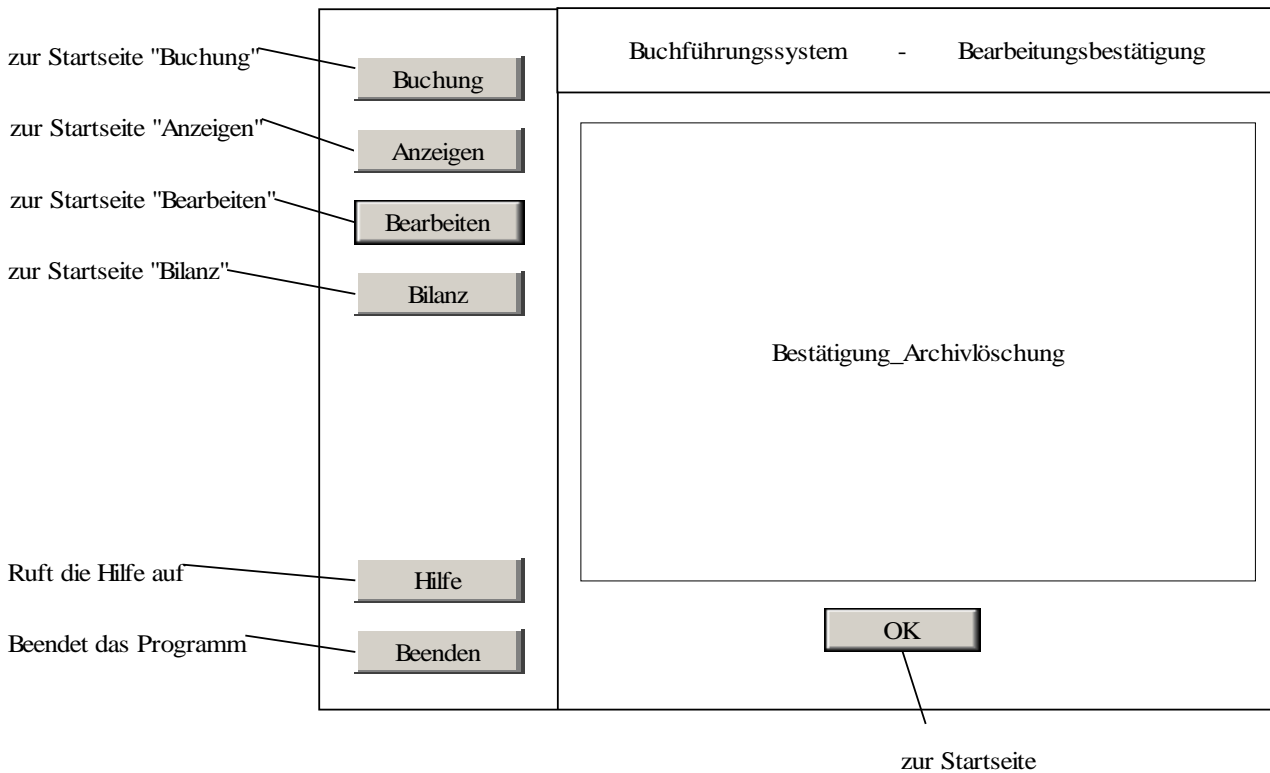
1.3.2.19 Fenster Bestätigung_Geschäftsfall_löschen



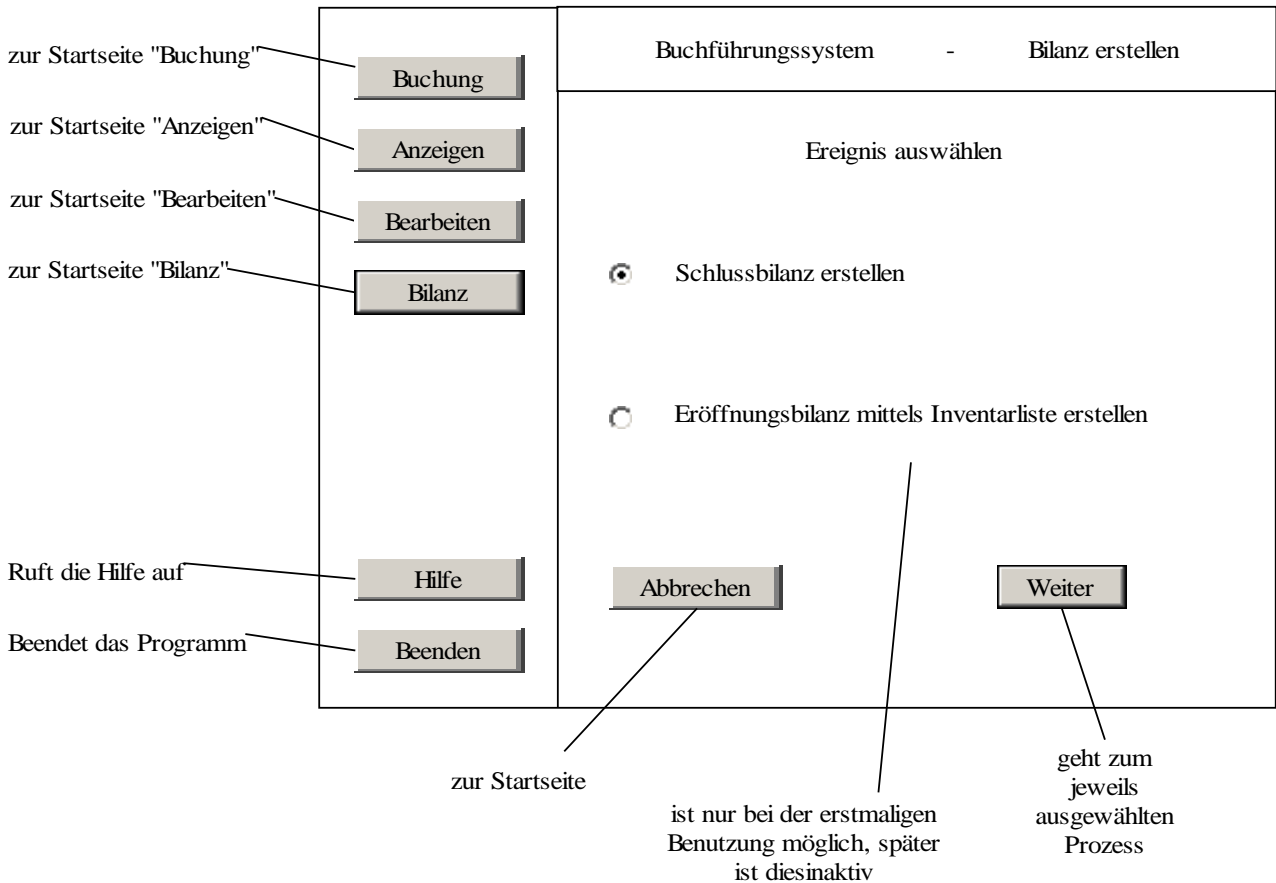
1.3.2.20 Fenster Archivdaten_löschen



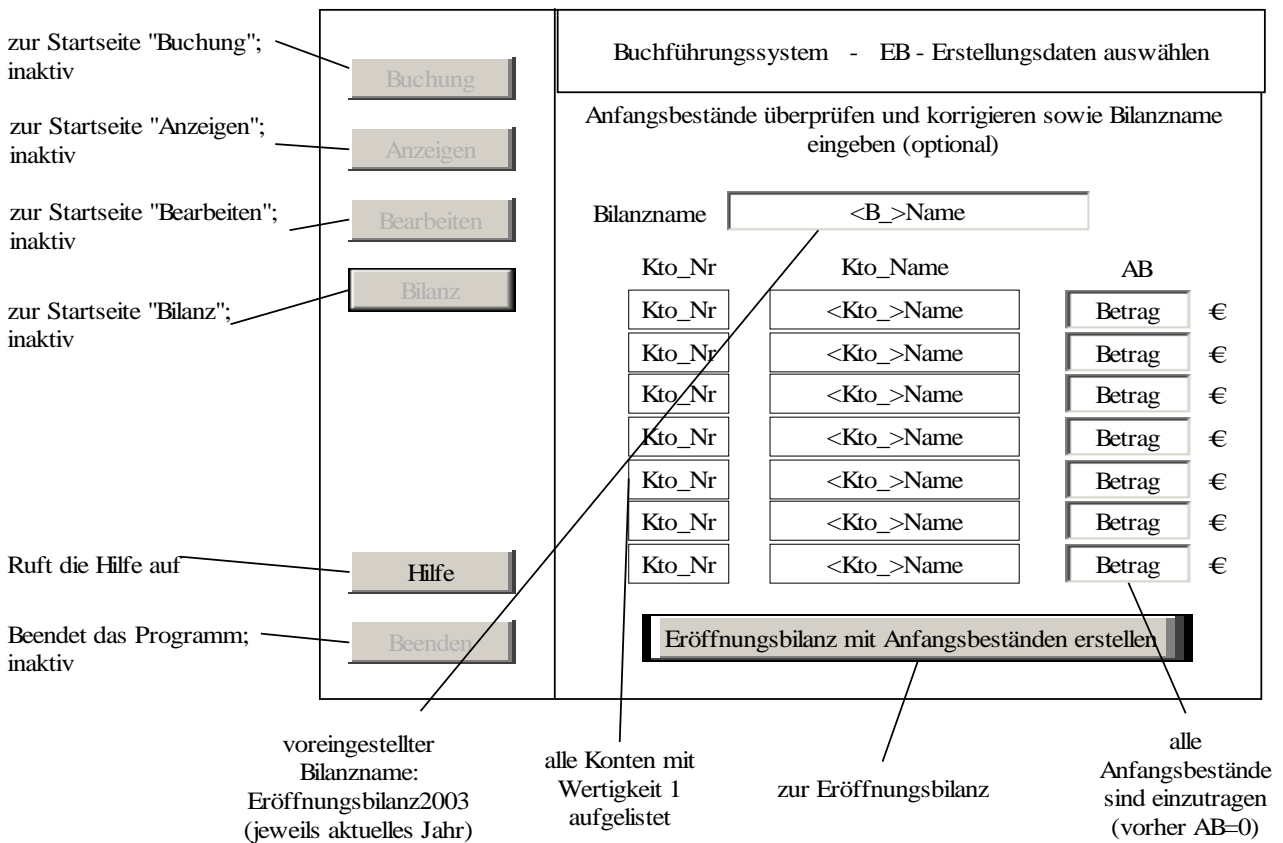
1.3.2.21 Fenster Bestätigung_Archivlöschung



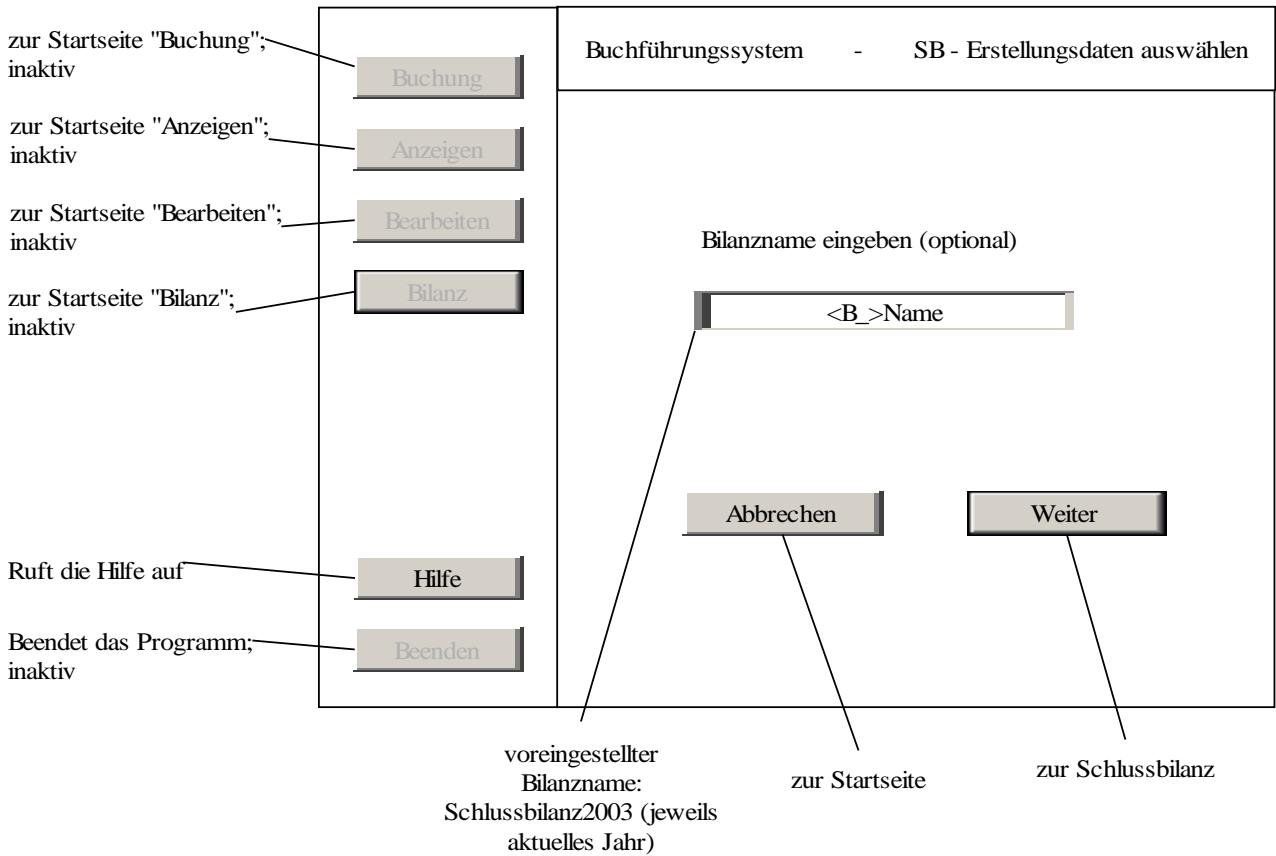
1.3.2.22 Fenster Bilanz



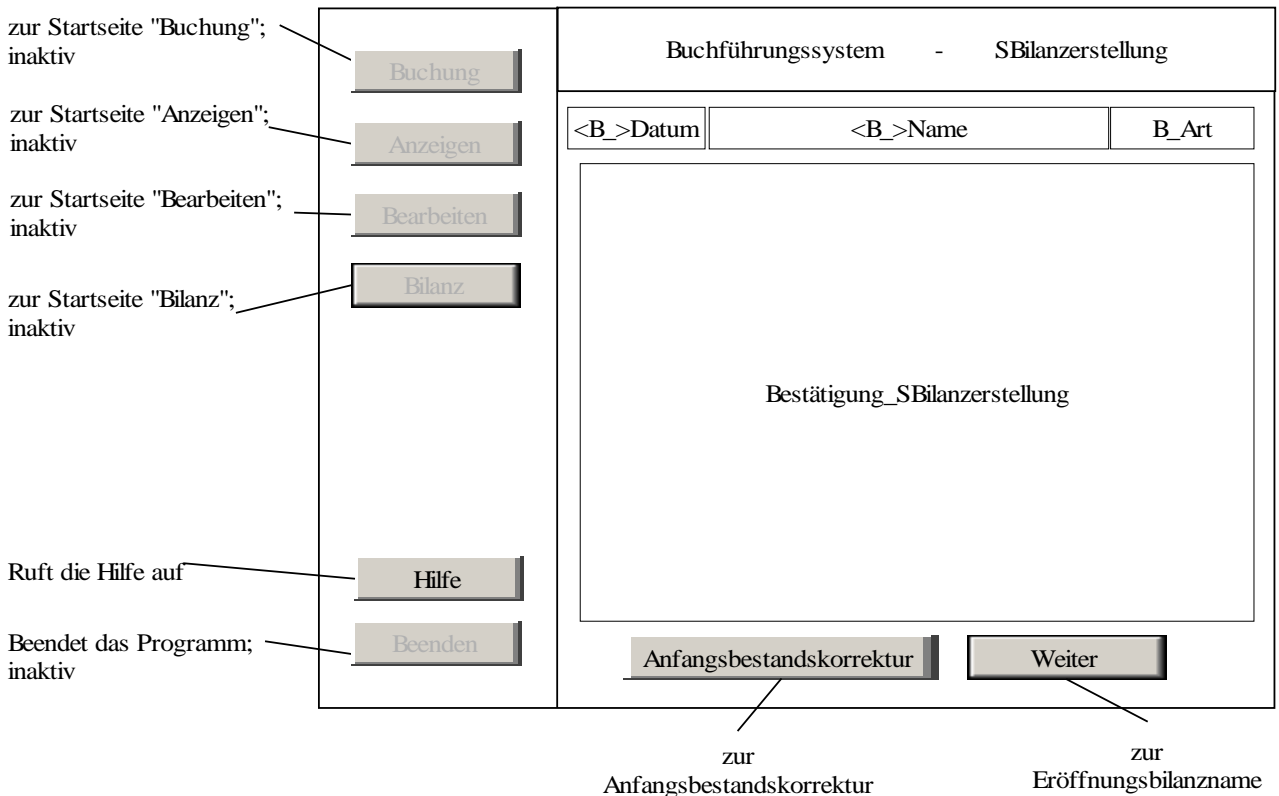
1.3.2.23 Fenster Eröffnungsbilanz_mit_Inventarliste



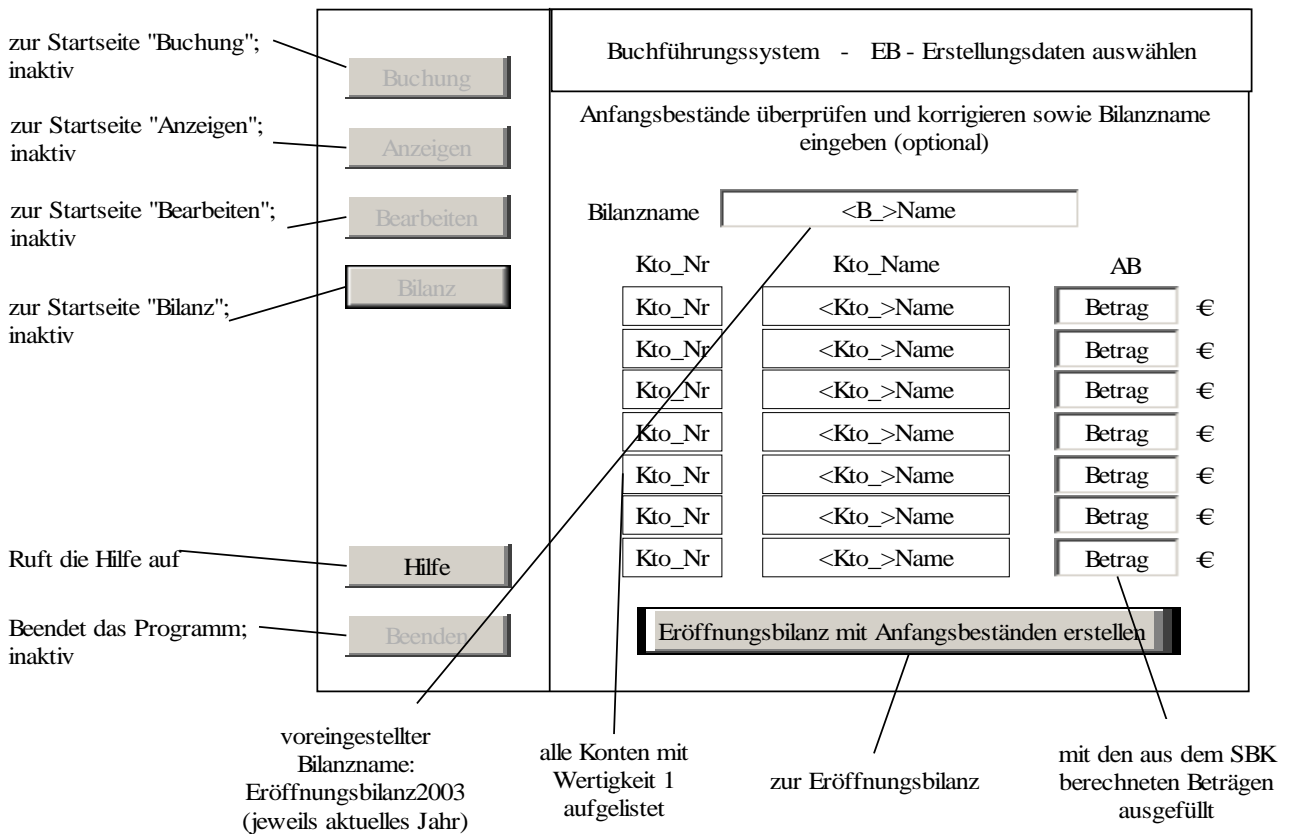
1.3.2.24 Fenster Schlussbilanzname



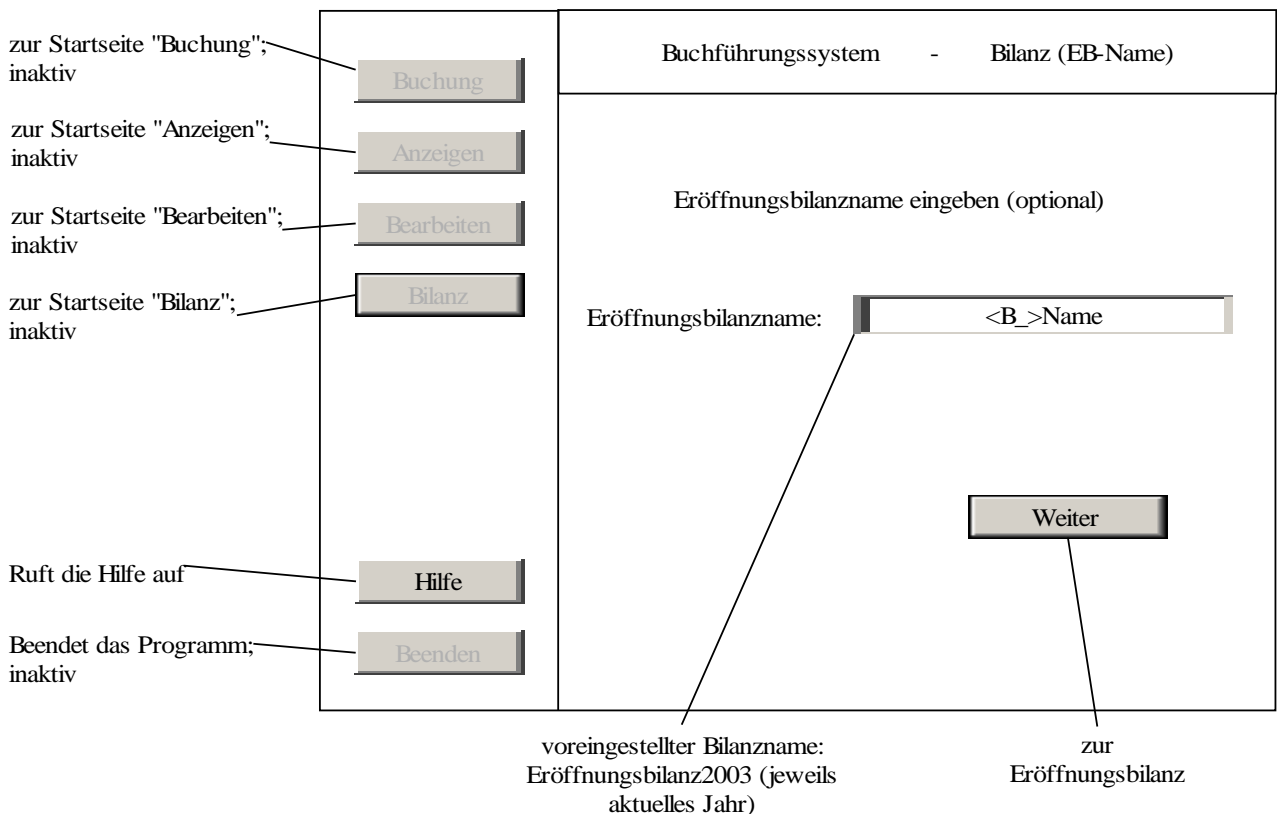
1.3.2.25 Fenster Schlussbilanz



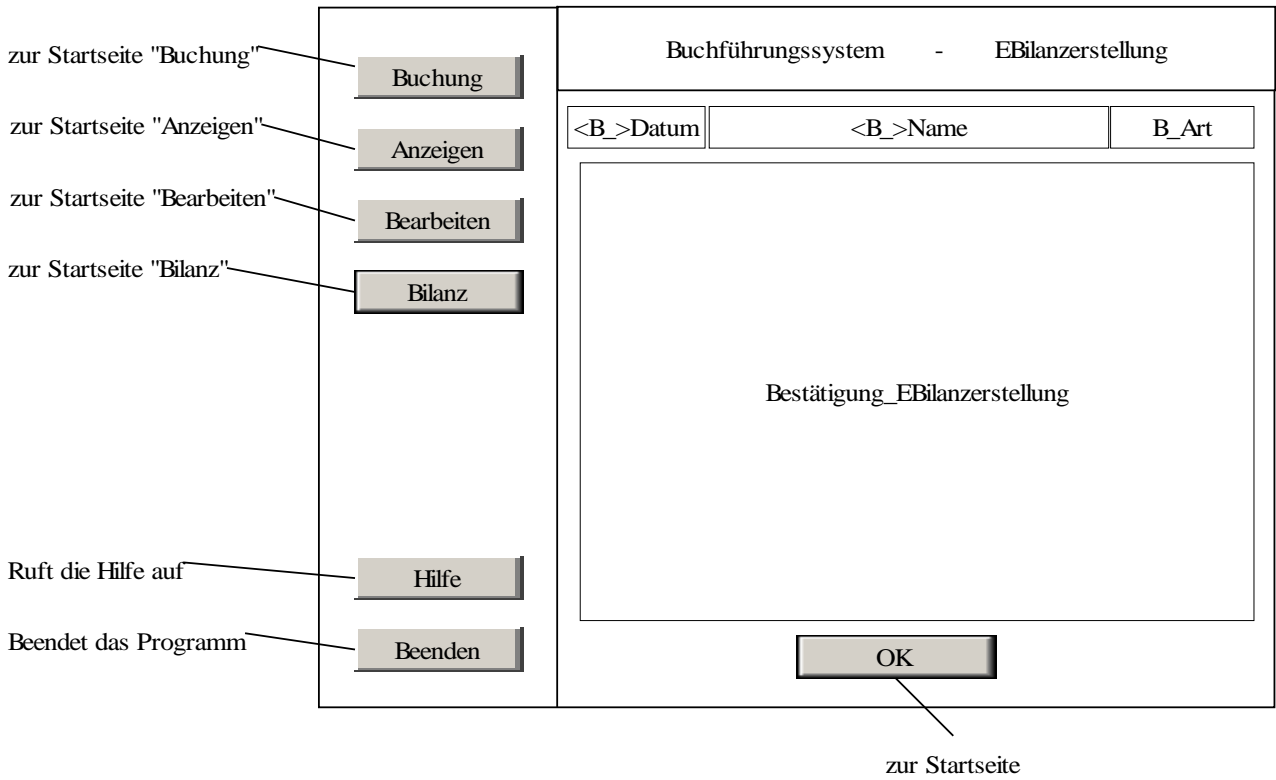
1.3.2.26 Fenster AB-Korrektur



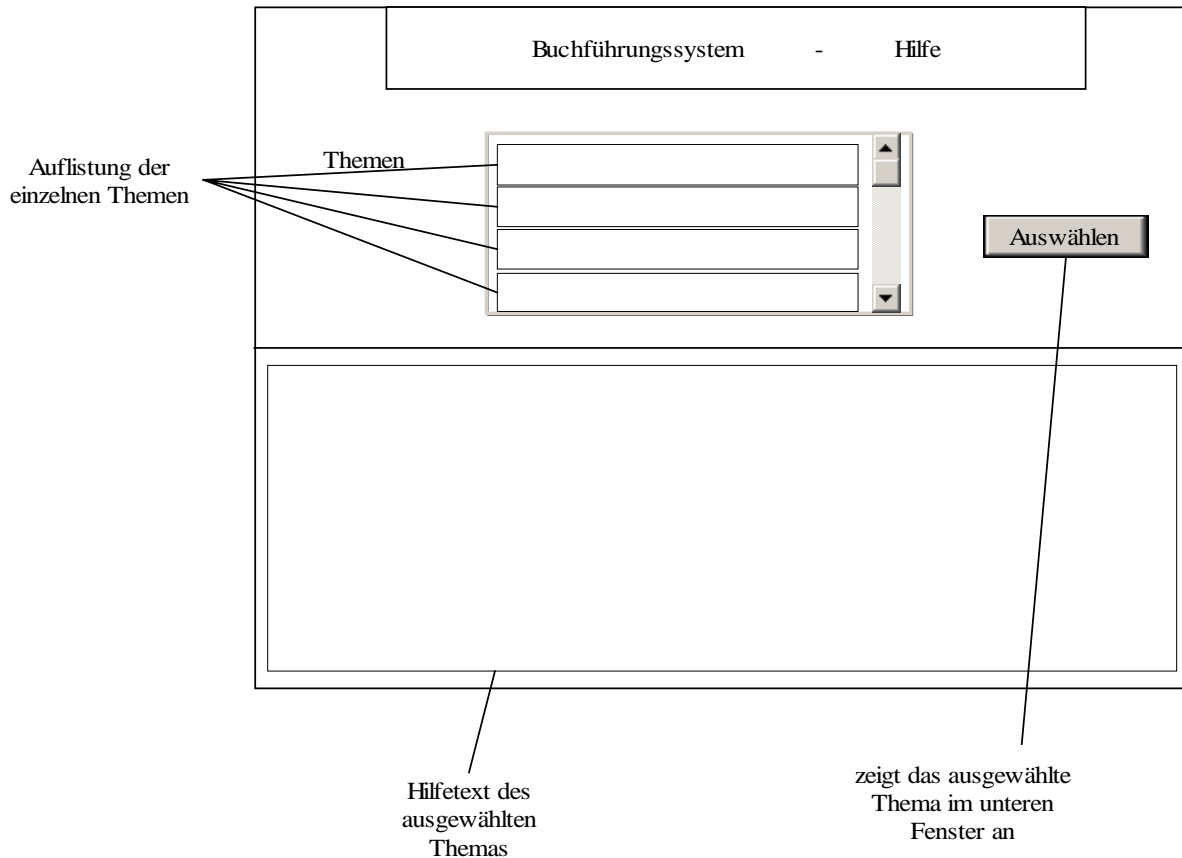
1.3.2.27 Fenster Eröffnungsbilanz_Name



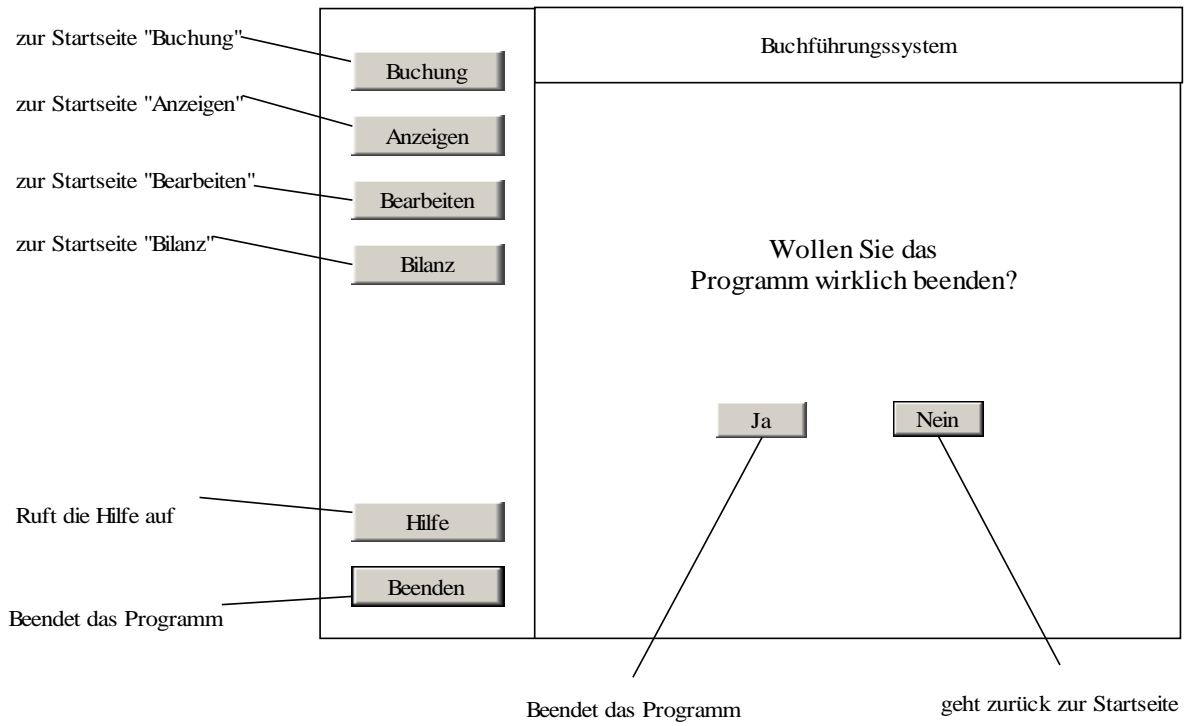
1.3.2.28 Fenster Eröffnungsbilanz



1.3.2.29 Fenster Hilfe



1.3.2.30 Fenster Beenden



Erläuterungen:

A :	Anzeigen	En :	Enter
AB :	AB-Korrektur	G :	mit GF buchen
Al :	Archiv löschen	GE :	Grundbuch Einsicht
BAL :	Bestätigung-Archivlöschung	GF :	GF anlegen
Be :	Bearbeiten	GFE :	GF Einsicht
BGl :	Bestätigung-GF_löschen	Gl :	GF löschen
Bi :	Bilanz	H :	Hilfe
BiE :	Bilanz Einsicht	K :	Konto anlegen
Bu :	Buchung	KA :	Konto auswählen
D :	Direkt buchen	N :	Nicht löschen
E :	Ende	ok :	zurück zur Startseite
EBN :	Eröffnungsbilanzname	S :	Schlussbilanz
EmI :	EB mit Inventarliste	SN :	Schlussbilanzname

Es kann außerdem von jedem Fenster aus die Hilfe aufgerufen werden. Außer bei den Unterfenstern der Bilanz, kann ebenfalls von jedem Fenster auf die Hauptpunkte (Buchung, Anzeigen, Bearbeiten, Bilanz, Beenden) zugegriffen werden und ein Rücksprung auf die Startseite ist möglich. Aus Überichtsgründen wurde dies alles jedoch weggelassen.

2 Spezifikation der operationellen Anforderungen

2.1 Operationelle Anforderungen an die Daten und die Datenbasen

Zur Abschätzung des Speicherplatzbedarfs und Datenaufkommens werden unter Bezug auf gängige Formate von MySQL folgende Annahmen gemacht:

- ganze Zahlen 1- und 2-stellig : 1 Byte
- ganze Zahlen 3- und 4-stellig : 2 Byte
- Beträge im Format: 10 Stellen, 2 Nachkommastellen und Vorzeichen: 14 Byte [Decimal(10,2)]
- für kurze Zeichenketten (im Allgemeinen kleiner 25 Zeichen) konstante Zeichenketten, d.h. 1 Byte pro Zeichen
- für längere Zeichenketten (Beschreibungen, (z.B. Konto)-Namen, etc.) variable Zeichenketten mit bis zu 255 Zeichen plus einem Zeichen für die Längenangabe. Entsprechend wurde in der unten stehenden Tabelle bei variablen Zeichenketten jeweils die maximal mögliche Länge benutzt/berechnet.

Für einige Strukturen lassen sich keine genauen Daten-/Speichergrößen angeben, da sie aus einer variablen Anzahl von Unterstrukturen bestehen. Deshalb wird teilweise darauf verzichtet, diese anzugeben bzw. es werden Mindestdatengrößen für (in der Regel) einen einfachen Datensatz angegeben. Bei den Fehlermeldungen wurde im Allgemeinen davon ausgegangen, dass kein Fehler auftrat und der Daten-/Speicheraufwand für diesen (in der Regel größeren) Teil abgeschätzt.

Element	Strukturbeschreibung	in Byte
Konten	{Konto}	{{21}+{21} + max 299}
Konto	@Kto_Nr + <Kto_>Name + Kto_Art + @<Kto_>Datumsflag + Kto_Wertigkeit + <Ober>Kto_Nr + Kto_Inhalt * SBK hat die vom System vergebene <Ober>Kto_Nr = 0000 *	2 + max 256 + 6 + 4 + 1 + 2 + {21}+{21}+28 = {21}+{21} + max 299
Kto_Nr	gZahl * 4stellig *	2
Kto_Art	["aktiv" "passiv" "SBK" "EK"]	6
Kto_Wertigkeit	gZahl * 1stellig, 0 <= Kto_Wertigkeit ; SBK hat Kto_Wertigkeit = 0, Hauptkonten 1, usw. *	1
Kto_Inhalt	<Soll_>Kto_Seite + <Haben_>Kto_Seite	{21} + {21} +28
Kto_Seite	Buchungseinträge + <Kto_Seiten>Summe	{21} + 14
Buchungseinträge	{Buchungseintrag}	{21}
Buchungseintrag	[(Beleg_ID) (<Gegen>Kto_Nr) + <GegenKto_>Name] + <Gegen_>Betrag * während des Geschäftsjahres wird Beleg_ID verwendet, <Gegen>Kto_Nr und <GegenKto_>Name nur bei Abschlussbuchungen und beim Vortragen des AB (dann <GegenKto_>Name = "AB" und <Gegen>Kto_Nr = 0000) *	[7 4 + 2] + 14 = max 21
Grundbücher	{Grundbuch}	{min 825}
Grundbuch	@<GB_>Datumsflag + GB_Inhalt	4 + {min 821}
GB_Inhalt	{Grundbucheintrag}	{min 821}
Grundbucheintrag	<Buchungs>Datum + Beleg_ID + Beleg_Kommentar + Kontenliste	4+7+ max 256 + min 554 = min 821
Beleg_ID	Datum + Zeit	4 + 3 = 7
Beleg_Kommentar	Zeichenkette80	max 256
Kontenliste	2{Kontenlisteneintrag}	2{277} = min 554
Kontenlisteneintrag	Buchungssatzzeile_blanko + <Buchungs>Betrag	263 + 14 = 277
Buchungssatzzeile_blanko	Kto_Nr + <Kto_>Name + betr_Kto_Seite	2 + max 256 + 5 = 263
betr_Kto_Seite	["Soll" "Haben"]	5
Bilanzen	{Bilanz}	
Bilanz	@<B_>Datum + <B_>Name + @B_Art + B_Inhalt	4 + max 256 + 2 + B_Inhalt = min 806
B_Art	["SB" "EB"]	2
B_Inhalt	<Aktiva_>B_Seite + <Passiva_>B_Seite	{272} + {272} + 28

Element	Strukturbeschreibung	in Byte
B_Seite	B_Seiteninhalt + <B_Seiten>Summe	{max 272} + 14
B_Seiteninhalt	{B_Posten}	{max 272}
B_Posten	BP_Nr + BP_Name + <BP_>Betrag	2 + max 256 + 14 = max 272
BP_Nr	Kto_Nr * Konten mit Kto_Wertigkeit = 1 treten in der Bilanz auf *	2
BP_Name	<Kto_>Name * Konten mit Kto_Wertigkeit = 1 treten in der Bilanz auf *	max 256
Geschäftsfälle	{Geschäftsfall}	{min 784}
Geschäftsfall	@GF_Nr + <GF_>Name + GF_Vorgangsdaten	2 + max 256 + min 526 = min 784
GF_Nr	gZahl * 3stellig *	2
GF_Vorgangsdaten	2{Buchungssatzzeile_blanko}	2{263} = min 526
Datumsflag	["-1" "0" <Abschluss>Datum] * Datumsflag = "-1" nur für Konten während des Kontenabschlussvorgangs; Datumsflag = "0" bei laufenden Konten und laufendem Grundbuch *	4
Summe	Betrag	14
Betrag	rZahl * 12stellig, davon 2 nach Komma + Vorzeichen; in EUR *	14
Datum	Tag + Monat + Jahr	1 + 1 + 2 = 4
Tag	gZahl * 2stellig, 01 <= Tag <= 31 *	1
Monat	gZahl * 2stellig, 01 <= Monat <= 12 *	1
Jahr	gZahl * 4stellig, 2000 <= Jahr <= 2100 *	2
Zeit	Stunde + Minute + Sekunde	1 + 1 + 1 = 3
Stunde	gZahl * 2stellig, 00 <= Stunde <= 23 *	1
Minute	gZahl * 2stellig, 00 <= Minute <= 59 *	1
Sekunde	gZahl * 2stellig, 00 <= Sekunde <= 59 *	1
Name	Zeichenkette	max 256
benannte DFD-Datenflüsse		
Kto-Änderungsdaten	["Konto anlegen" + Kto_Nr + <Kto_>Name + Kto_Art + (<Ober>Kto_Nr)] * optionale Teile der 1. Option entfallen nur beim SBK und bei den Hauptkonten *	[1 + 2 + max 256 + 2 + 2] = [max 263 3]
Kto-Änderungsauswahl	["Konto anlegen"] * 0 = „anlegen“	1
Bestätigung_Kto-Änderung	[Bestätigung_Kto_anlegen]	1
Bestätigung_Kto_anlegen	<Kto_anlegen>FehlerNr + (Konto) * optionaler Teil nur bei FehlerNr = 0 *	2 + (min 241)
Kto-Bezeichnung	<Kto_>Datumsflag + [Kto_Nr <Kto_>Name]	4 + [4 max 257] = max 261
Kontoinformationen	Konto	min 241
GB-Datum	<GB_>Datumsflag	4
Grundbuchinformationen	Grundbuch	4
GF-Änderungsdaten	[GF-Änderungsauswahl + Geschäftsfall GF-Änderungsauswahl + GF_Nr + (<GF_>Name)]	[1 + min 784 1 + max 256] = [min 785 max 257]
GF-Änderungsauswahl	["Geschäftsfall anlegen" "Geschäftsfall löschen"] * 0 = „anlegen“, 1 = „löschen“ *	1
Bestätigung_GF-Änderung	[Bestätigung_GF_anlegen Bestätigung_GF_löschen]	1
Bestätigung_GF_anlegen	<GF_anlegen>FehlerNr + (Geschäftsfall) * optionaler Teil nur bei FehlerNr = 0 *	2 + (min 784)
Bestätigung_GF_löschen	<GF_löschen>FehlerNr + ("Geschäftsfall wurde gelöscht.") * optionaler Teil nur bei FehlerNr = 0 *	2 + max 256
GF-Bezeichnung	GF_Nr + (<GF_>Name)	2 + max 256 = max 258
Geschäftsfallinformationen	Geschäftsfall	min 784
Buchungsinformationen	<Buchungs>Datum + Beleg_Kommentar + ["direkt" + Kontenlisten "über Geschäftsfall" + GF_Nr + Buchungsbeträge] * die Kardinalität von Buchungsbeträge entspricht der Kardinalität der im ausgewählten Geschäftsfall angelegten {Buchungssatzzeile_blanko} *	4 + max 256 + [1 + min 554 1 + 2 + min 28] = max 260 + [min 555 min 31]
Buchungsauswahl	["direkt" "über Geschäftsfall"] * 1 = direkt, 2 = über GF*	1
Bestätigung_Buchung	<Buchung>FehlerNr + ("Buchungssatz unter Belegnummer " + Beleg_ID + " erfolgreich gebucht.") * optionaler Teil nur bei FehlerNr = 0 *	2 + max 256 + 7 = max 265
AB-Änderungsdaten	AB-Änderungsliste + <B_>Name	= { max 272 } + max 256
AB-Änderungsliste	{AB-Änderungslisteneintrag}	{ max 272 }
AB-Änderungslisteneintrag	Kto_Nr + (<Kto_>Name) + <AB_>Betrag	2 + max 256 + 14 = max 272
Bestätigung_EBilanzerstellung	<EBilanzerstellung>FehlerNr + (Bilanz) * optionaler Teil nur bei FehlerNr = 0 *	2 + min 806 = min 808
SBilanz-Erstellungsauftrag	<B_>Name	max 256

Element	Strukturbeschreibung	in Byte
Bestätigung_SBilanzerstellung	<SBilanzerstellung>FehlerNr + (Bilanz) * optionaler Teil nur bei FehlerNr = 0 *	2 + min 806 = min 808
Bilanz-Bezeichnung	<B_>Datum + (<B_>Name) + B_Art	4 + max 256 + 2 = 262
Bilanzinformationen	Bilanz	min 806
Archiv-Löschungsauftrag	<A_>Datum	4
Bestätigung_Archivlöschung	<A_Daten_löschen>FehlerNr + ("Die Datenbestände wurden gelöscht.") * optionaler Teil nur bei FehlerNr = 0 *	4 + max 256 = max 260
FehlerNr	gZahl * 4stellig, repräsentiert den informierenden Text aus der Fehlercodetabelle *	2
benannte ERD-Elemente		
trägt_ein_in	@GF_Nr + <Buchungs>Datum + Beleg_Kommentar + Beleg_ID + Buchungsbeträge + <GB_>Datumsflag * <GB_>Datumsflag = 0 *	2 + 4 + max 256 + 7 + min 28 + 4 = min 297
bucht_auf	@GF_Nr + Beleg_ID + Buchungsbeträge + 2{Kto_Nr} * Buchungsbeträge und {Kto_Nr} treten mit gleicher Kardinalität auf *	2 + 7 + min 28 + min 4 = min 41
eröffnet	@<B_>Datum + B_Art + [@Kto_Nr <GB_>Datumsflag]	4 + 2 + [2 4] = max 10
abgeschlossen_zu	@Kto_Nr + <B_>Datum + B_Art * B_Art = "SB" *	2 + 4 + 2 = 8
beendet	@<B_>Datum + B_Art + @<GB_>Datumsflag	4 + 2 + 4 = 10
Buchungsbeträge	2{<Buchungs>Betrag}	2{14} = min 28

2.1.1 Speicher (Angaben zum Speicherbedarf in Byte)

Speicher	Struktur	Minimum	Mittel	Maximum
Geschäftsfälle	{Geschäftsfall}	0 * 784	50 * 784 = 32,2 kB	100 * 784 = 78,4 kB
Konten	{Konto}	0 * 241	10 * (56 + 299) * 12mon * 10a * = 426kB	30 (56 + 299) * 12mon * 10a = 1,278MB
Konten gesamt	{Konten}	0 * 241	50 Konten * 426 kB = 21,3MB	100 Konten pro Jahr * 1,278MB = 127,8MB
Grundbücher	{Grundbuch}	0 * 825	10 Buchungen pro Konto bei 50 Konten: ca. 25MB	100 Buchungen pro Konto bei 100 Konten: ca. 140MB
max. Gesamtbedarf				ca. 270MB

2.1.2 Weitere Datenflüsse (Angaben in Byte)

Datenfluss	Struktur	Richtung	Minimum	Mittel	Maximum
Kontoinformationen	{Konto}	Server -> Client	0 * 784	32,2kB	78,4kB
Fehlermeldungen pro Fehler/ Bestätigungen	{FehlerNr + Fehlername}	Server -> Client			max. 258
Kto-Änderungsdaten					max. 263
Bilanzinformationen	{Bilanz}	Server -> Client	806	ca 28kB	ca. 55kB
Buchung / GF-anlegen		Client -> Server			ca. 300
Verschiedene Auswahlen treffen		Client -> Server	1	20	40

2.1.3 Abschätzungen

- Gesamtsystem: Es wird von einer erwarteten Nutzungsdauer von mindestens 10 Jahren ausgegangen (Benutzung der Angabe ‚a‘ in den Tabellen oben). Es wird davon ausgegangen, dass alte Datenbestände (gesicherte/archivierte Konten) nach 10 Jahren manuell gelöscht werden, so dass sämtliche gespeicherte Datenbestände normalerweise 10 Jahre alt sind und damit das durchschnittliche gespeicherte Datenvolumen nach oben begrenzt ist. Weiterhin wird von einem mittelständischen Unternehmen ausgegangen, welches dieses System benutzt.
- Konten: Es wird angenommen, dass im Schnitt 50, maximal 100 Konten pro Jahr benutzt/angelegt werden.
- Buchungen: Es wird davon ausgegangen, dass im Mittel 10 und maximal 30 Buchungen pro Konto und pro Monat durchgeführt werden.
- Geschäftsfälle: Die Anzahl der Geschäftsfälle wird auf 100 begrenzt, im Mittel werden 50 benutzt.

2.1.4 Genauigkeit

Es werden keine erhöhten Anforderungen an die Genauigkeit gestellt. Beträge werden mit maximal 10 Stellen, einer Vorzeichenstelle und zwei Nachkommastellen gespeichert und verarbeitet. Dabei werden ganze Zahlen verwendet, so dass keine Rundungsfehler bei den (am Häufigsten vorkommenden) arithmetischen Funktionen (Addition und Subtraktion) auftreten können.

2.2 Operationelle Anforderungen an die Datenströme

Während der Eingabe von Buchungen, dem Anlegen von Konten und Geschäftsfällen, sowie der Änderung von Geschäftsfällen treten nur sehr kleine Datenströme vom Client (Benutzer) zum Server auf. Größere Datenströme entstehen nur beim Anzeigen von Konten und beim Anzeigen von Bilanzen. Die Berechnungen zur Bilanzerstellung erfolgen dabei auf dem Server, so dass dieser eine höhere Belastung erfährt, der Clientrechner jedoch keine große Belastung hat, da er lediglich die bereits fertig verarbeiteten Daten empfängt und anzeigt. Während der Bilanzerstellung können größere Datenströme sowie eine längere Bearbeitungszeit des Servers auftreten.

2.3 Operationelle Anforderungen an die Prozesse

Im Normalfall werden Antwortzeiten von bis zu 5 Sekunden auftreten. Bei der Bilanzerstellung und der Kontoinhaltsanzeige treten größere Datenmengen auf, deren Verarbeitung länger als 5 Sekunden in Anspruch nehmen kann, wobei der Nutzer vor Ausführung der Funktion darauf hingewiesen wird, dass es voraussichtlich eine längere Zeit dauern kann, bis alle Daten vom System verarbeitet sowie anschließend zum Client übertragen worden sind.

3 Spezifikation wichtiger Qualitätsanforderungen

3.1 Benutzbarkeit

Die Benutzer des Systems sind mit den Vorgängen der Buchführung bestens vertraut, nicht aber unbedingt mit den technischen Realisierungsmöglichkeiten von Software. Daher müssen sich die vom System angebotenen Nutzungsmöglichkeiten dem Fachwissen des Nutzers anpassen und ihm einen intuitiven Zugang zur Bedienung gewährleisten.

Dies soll erreicht werden, indem die ausführbaren Vorgänge den in der Buchführungspraxis gängigen Verfahrensweisen entsprechen, die Benutzerführung konsistent ist, d.h. an keiner Stelle Unklarheiten über den nächsten auszuführenden Schritt bietet und darüber hinaus den Anwender von technischen und organisatorischen Fragen der Datenverwaltung isoliert.

Außerdem soll an jeder Stelle des Programms eine Online-Hilfe verfügbar sein, die Aufschluss über Funktionalität und Bedienung des Systems gibt.

3.2 Zuverlässigkeit

Benutzereingaben, die über die Tastatur erfolgen, müssen grundsätzlich auf ihre Schlüssigkeit in Hinsicht auf Typ und Format geprüft werden. Dort, wo durch den Programmierer automatische Konvertierungen vorgesehen werden – beispielsweise von Text nach Währungsformaten – ist durch einen Dialog mit dem Nutzer sicherzustellen, dass die aus der Konvertierung hervorgegangenen Werte dessen Intentionen entsprechen.

Hinsichtlich der mathematischen Operationen sind Überlegungen anzustellen, wo durch Benutzung von Gleitkommaarithmetik Rundungs- bzw. Darstellungsfehler (insbesondere bei Multiplikation und Division) auftreten können, welche die Zahlbeträge verfälscht, so dass letztendlich Fehlbeträge in Bilanzen sichtbar würden. Diesen Fehlerquellen ist besondere Aufmerksamkeit zu widmen und es sind Verfahren zu verwenden, die dies weitgehend ausschließt.

Die Konsistenz der Datenbank muss dadurch gewährleistet werden, dass erst, wenn alle Informationen zum entsprechenden Vorgang gesammelt wurden, der gesamte Datensatz gespeichert wird, wobei über diesen Zeitraum der Datenspeicher für andere konkurrierende Schreibwünsche gesperrt wird. Über einen misslungenen Schreibversuch ist der Benutzer zu informieren, damit er ihn zu einem späteren Zeitpunkt wiederholen oder den Vorgang abrechnen kann.

3.3 Integrität

Die Daten werden zentral auf einem MySQL Server gehalten, der seitens der Unternehmensleitung natürlich besonderen Schutzes bedarf. Die Datenbank sollte durch Benutzernamen und Passwörter vor unbefugter Datenmanipulation oder Dateneinsicht geschützt werden. Die Datenkonsistenz für Multisession-Betrieb wird gewährleistet. Darüberhinaus sind sämtliche Webseiten des Systems durch eine geeignete Zugriffskontrolle (z.B. .htaccess) vor unauthorisiertem Zugriff zu schützen.

3.4 Flexibilität

Das Programm stellt eine funktionale Grundstruktur zur Verfügung. Die Anpassung der Kontenstruktur, der Vorlagen und dazugehörigen Menüs werden vom Nutzer vorgenommen.

3.5 Portabilität

Besondere Maßnahmen zur Portabilität müssen nicht getroffen werden, da der technische Entwurf weitgehend von einer bestimmten Rechner- oder Betriebssystemstruktur abstrahiert.

Die notwendigen Server sind für nahezu alle Plattformen verfügbar und die Anforderungen für den Anschluss neuer Arbeitsstationen sind denkbar gering, da ein handelsüblicher PC mit HTML-fähigen Browser und Verbindung zum zentralen Rechner für den Zugang ausreichend ist.

4 Basismaschine und Entwicklungsumgebung

4.1 Basismaschine

In den nachfolgenden Spezifikationen wird jeweils zwischen dem Client (Nutzerschnittstelle) und dem Server unterschieden.

4.1.1 Hardwarekonfiguration

Bezüglich der Hardware wird folgende Hardwareausstattung als erforderlich erachtet:

Client:

- Basis-PC (ab Pentium) mit Tastatur, Maus, Monitor (min. 256 Farben, Auflösung von 1024x768)
- A4-Drucker (optional)
- Schnittstelle für Netzwerk

Server:

- Basis-PC (ab Pentium)
- Festplatte mit einer freien Kapazität von min. 500 MB
- Schnittstelle für Netzwerk

Die Geschwindigkeit der abzuarbeitenden Prozesse wird hauptsächlich von der Hardwarekonfiguration des Servers bestimmt. Der Client-PC hat auf die Gesamtperformance nur minimalen Einfluss.

4.1.2 Betriebssystem

Als Betriebssystem wird vorgesehen:

Client:

- Beliebig

Server:

- Beliebig (bevorzugt UNIX)

4.1.3 Sonstige Basissoftware

Benötigte installierte Software:

Client:

- Beliebiger Webbrowser mit graphischer Oberfläche

Server:

- Webserver (bevorzugt Apache) mit PHP4-Support und MySQL-Schnittstelle
- MySQL-Datenbank-Server

4.2 Entwicklungsumgebung

Die Umsetzung des Projekts erfolgt mithilfe der Skriptsprache PHP4 und als Darstellungssprache HTML. Da es hierfür keine richtige Entwicklungsumgebung gibt, wird alles mit einem normalen Text-Editor umgesetzt.

5 Planung

