

Fachhochschule der Wirtschaft
-FHDW-
Paderborn

Projektarbeit

Thema:

**Analyse, Konzept und Realisierung eines Auftragserfassungssystems für
den steuerberatenden Beruf auf der Basis von MS-Access**

Prüfer:

Herr Prof. Dr. Carsten Weigand
Herr Prof. Dr. Werner Oertmann

Verfasser:

Daniel Becker
Im Obstgarten 5
32657 Lemgo

4. Hochschulquartal
Studiengang Wirtschaftsinformatik

Eingereicht am:

10. April 2000

Inhaltsverzeichnis	Seite
Abbildungsverzeichnis	II
Abkürzungsverzeichnis	III
Glossar	IV
1 Einleitung	1
1.1 Unternehmensportrait.....	1
1.2 Problemdarstellung	1
1.3 Zielsetzung und Gang der Arbeit	3
2 Planung der Entwicklung	3
2.1 Bisheriger organisatorischer Ablauf in der Kanzlei.....	3
2.1.1 Die Neumandatsanlage	4
2.1.2 Die Auftragerfassung und -änderung	5
2.2 Das Pflichtenheft.....	5
2.3 Wahl der Programmierumgebung.....	6
2.4 Netzwerkfähigkeit und -einsatz des AES.....	7
3 Datenbestand	8
3.1 Benötigte Daten.....	8
3.2 Datenherkunft.....	9
3.3 ER-Modell der Daten	10
3.4 Transformationsprozeß und Normalisierung	12
3.5 Erstellung von Abfragen unter Access.....	16
3.6 ZMSD-Datenabgleich	16
4 Benutzerdialoge	19
4.1 Hierarchische Funktionsgliederung des Programmes.....	19
4.2 Gestaltung der Forms	20
4.2.1 Probleme des Multithreadings unter Access	21
4.2.2 Aufbau eines Dialoges am Beispiel der Auftragsänderung	24
4.3 Drucken im AES	26
5 Testphase	29
6 Zusammenfassung und Ausblick	31
Quellenverzeichnis	33
Anhangsverzeichnis	34
Ehrenwörtliche Erklärung	37

Abbildungsverzeichnis

Abbildung	Seite
Abbildung 1: Auszug aus dem Tätigkeitsplan für die LEA-Erfassung	2
Abbildung 2: Die Entity-Typen mit ihren Attributen	10
Abbildung 3: Die Relationships	11
Abbildung 4: Das Relationenmodell des AES	14
Abbildung 5: Darstellung der Beziehungen unter Access	15
Abbildung 6: Alle nicht gedruckten Aufträge ermitteln	16
Abbildung 7: Hierarchische Funktionsgliederung des AES	19
Abbildung 8: Codeauszug „Nummer des aufrufenden Forms ermitteln“	23
Abbildung 9: Beispielformular „Auftragsänderung“	24
Abbildung 10: Das Druckmenü	27
Abbildung 11: Codeauszug „Alle Aufträge drucken“	28
Abbildung 12: Beispiel einer von Access automatisch erzeugten Fehlermeldung	30

Abkürzungsverzeichnis

AES	Auftragserfassungssystem
DBMS	Datenbankmanagementsystem
dtsch.	deutsch
engl.	englisch
etc.	et cetera
f.	folgend
ff.	fortfolgend
ggf.	gegebenfalls
IS	Informationssystem
s.	siehe
S.	Seite
s.a.	siehe auch
sog.	sogenannt
StBGebV	Steuerberater Gebührenverordnung
u.a.	unter anderem
usw.	und so weiter
u.U.	unter Umständen
VB	Visual Basic
vgl.	vergleiche
z.B.	zum Beispiel

Abkürzungen der Schlüsselattribute:

A_Key:	Primärschlüssel der Auftragsarten
G_Key:	Primärschlüssel der Gegenstandswerte
M_Key:	Primärschlüssel der Mandantendaten

Glossar

DATEV: Rechenzentrum für den steuerberatenden Beruf mit Sitz in Nürnberg und einer Zweigniederlassung in Hannover. Die DATEV betreut bundesweit steuerberatende Kanzleien und versorgt diese mit Erfassungs-, Auswertungs- und Controllingsoftware.

Drag-and-Drop: Name einer Eingabetechnik, die durch die Maus möglich wurde. Der Benutzer kann ein Objekt mit der Maus auswählen (drag), über eine andere Anwendung ziehen und fallen lassen (drop).

ERM, ER-Modell: Kurzform für „Entity-Relationship-Modell“. Das ERM stellt eine Hilfe für die Entwicklung von Programmen und Datenbanken im Speziellen dar, um die reale Welt schematisch abzubilden. Dabei geht das ERM davon aus, „daß sich die zu modellierende „Welt“ aus genau abgrenzbaren, individuellen Exemplaren von Dingen, Personen, Begriffen usw. (engl. *entities*) zusammensetzt und daß zwischen diesen Exemplaren Beziehungen (engl. *relationships*) bestehen“¹.

Integrität, referentielle²: Unter Access können Beziehungen zwischen den Schlüsselementen der Tabellen mit der sog. referentiellen Integrität eingerichtet werden. Ist sie aktiviert, sorgt dann Access selbst dafür, daß z.B. auch alle Aufträge eines Mandanten gelöscht werden, wenn der Mandant selbst gelöscht wird. Genauso ist es möglich, das Löschen eines Mandanten zu verbieten, sollten noch Aufträge zu dem Mandat existieren. Die Beziehungsdarstellung unter Access hat jedoch wenig mit dem ERM zu tun und stellt keinen Ersatz dafür dar.

LEA: Leistungserfassung und -abrechnung

Stellt einen Datensatz dar, bestehend aus Auftragsart, Tätigkeit, Abrechnungsdatum, Mandantenummer, Dauer, Datum und ggf. einer Notiz.

Siehe auch Abbildung 1.

Medienbruch: Als „Medienbruch“ bezeichnet man alle Vorgänge, die bereits digital erfaßte Daten analog speichern, also z.B. ausdrucken, und diese analog gespeicherten Daten nachträglich wieder in das digitale Format gebracht werden, z.B. durch Abtippen oder Einscannen. Wird der Medienbruch in den betrieblichen Prozessen eines Unternehmens häufig praktiziert oder durch schlecht organisierte Abläufe nötig, so schleichen sich oft bei der Umwandlung der Daten Fehler ein, die sich sonst hätten vermeiden lassen. Daher ist es wünschenswert, daß Daten so lange wie möglich in der ursprünglichen digitalen Form bleiben und auch in dieser Form bearbeitet werden.

Multithreading: Zu deutsch: „nebenläufige Verarbeitungsschritte“ (engl. *threads*)³. In diesem Zusammenhang ist damit gemeint, daß die VB-Module eines jeden Formulars nebeneinander ablaufen und nach dem Befehl zum Aufruf eines weiteren Formulars nicht auf dessen Beendigung warten, wie es ein prozeduraler Aufbau erwarten ließe.

¹ DUDEN Informatik, 1993, S. 231, Entity-Relationship-Modell

² vgl. Access Hilfe, 1997, Datei „ACMAIN80.HLP“, Stichwort „Referentielle Integrität“

³ vgl. Hansen, 1998, S. 878f.

Glossar (Fortsetzung)

PC-KORG: Kanzleiorganisation (DOS-Version)

Programm zur Rechnungserstellung auf Basis der erfaßten LEA-Sätze eines jeden Mitarbeiters.

SQL: Abkürzung für „Structured Query Language“ (dtsch.: „Strukturierte Abfrage Sprache“). Diese für DBM-Systeme eingeführte Sprache der vierten Generation wird für relationale Datenbanken verwendet. Sie erlaubt das Suchen bzw. Auswählen (select), das Verändern (update) und Löschen (delete) von Datensätzen auf der Basis von Relationen. Ein einfaches Beispiel für eine SQL-Anweisung ist die Select-Anweisung, die folgendermaßen aussehen kann: „SELECT x,y,z FROM relation WHERE x=y“⁴

WYSIWYG: Abkürzung für „What you see is what you get“. Ein Begriff, der durch Textverarbeitungen auf graphischen Systemen geprägt wurde. Gemeint ist, daß der Anwender bereits vor dem Druck sehen kann, wie sein Dokument auf dem Papier aussehen wird.⁵

ZMSD: Zentrale Mandanten-Stammdaten

In diesem Programm werden alle Daten, wie Adressdaten, Bankverbindungsdaten und ähnliches, eines Mandanten erfaßt und verwaltet.

⁴ vgl. Hansen, H.R., 1998, S. 851 und S. 957ff., sowie DUDEN Informatik, 1993, S. 684ff.

⁵ vgl. Hansen, H.R., 1998, S.185

1 Einleitung

1.1 Unternehmensportrait

Das Steuerberatungsbüro Winfried Becker wurde von dem Steuerberater, vereidigten Buchprüfer und Rechtsbeistand Winfried Becker 1977 aus der Buchstelle der Kreishandwerkerschaft Lippe gegründet. Mit fünfundzwanzig Mitarbeiter werden ca. 600 Mandate betreut, welche sich größtenteils aus Klein- und Mittelbetrieben des Handwerks zusammensetzen.

Zum Aufgabengebiet der Kanzlei gehört die Anfertigung der Finanz- und Lohnbuchhaltung mit sämtlichen Anmeldungen und Voranmeldungen sowie der Erstellung der Jahresabschlüsse und der entsprechenden Steuererklärungen für die Finanzverwaltung. Ergänzt wird der Aufgabenbereich durch die steuerliche- und betriebswirtschaftliche Beratung, sowie der Nachfolgeberatung.

Zur Erfüllung dieser Aufgaben ist die Steuerkanzlei Mitglied der DATEV und verwendet neben einiger Fremdsoftware, wie dem Office-Paket von Microsoft, beinahe alle Softwareprodukte der DATEV. Dazu werden ein NT-Server als Fileserver, ein Novell-Server als Printserver, eine NT-Workstation als Kommunikationsserver sowie 25 Arbeitsplatz PCs mit dem Betriebssystem Windows 95 und 98SE unter strukturierter Netzverkabelung eingesetzt.

1.2 Problemdarstellung

In einem Dienstleistungsunternehmen mit einem umfangreichen Leistungskatalog ist es unumgänglich, vor dem Beginn der Tätigkeiten den genauen Auftrag des Kunden vorliegen zu haben⁶. Dies ermöglicht die genaue und vor allem zeitnahe Abrechnung⁷ mit der notwendigen Transparenz für den Kunden. Durch den Zwang einer Auftragserteilung wird zudem der Kunde angehalten, sich genau mit den angebotenen Leistungen auseinanderzu

⁶ vgl. Barz, D. / Wunram, S., 1999, S. 71

⁷ vgl. Barz, D. / Wunram, S., 1999, S. 81

setzen und die für ihn zutreffenden auszuwählen⁸. Dadurch wird dem Mandanten der gesamte Umfang der notwendigen Arbeiten an seinem Auftrag klarer und nachvollziehbar.

Abbildung 1: Auszug aus dem Tätigkeitsplan für die LEA-Erfassung

Tätigkeitsplan für Kanzleiorganisation			
33	Buchführung (Jahr/Monat)	24	Private Steuern (Bearbeitungsjahr)
	3301 Buchführung		2401 Einkommensteuererklärung
			2411 VermSt (Privat)
			2416 Erbschaft- / Schenkungsteuer
34	Lohn (Jahr/Monat)	21	Steuerliche Beratung (aktuelles Jahr)
	3410 Lohnabrechnung		2101 *Steuerliche Beratung
			2302 *Anträge
			2901 *Mitwirkung bei Betriebsprüfung
			2801 *Steuerbescheidprüfung
			4101 *Rechtsbehelf
32	Sonderleistungen FIBU	50	Nichtsteuerliche Beratung (aktuelles Jahr)
	3201 Einrichten einer Buchführung		5000 *Wirtschaftliche Beratung
	3350 *Hilfeleistung bei der Buchführung		5005 *Vertragsberatung
			5010 *Finanzierung
	3331 EDV-Kosten FIBU		5015 *Kauf-/Verkaufsberatung
	3370 *Sonstiges FIBU		5020 *Kreditverhandlungen
			5040 *Wirtschaftliche Gestaltung
			5060 *Treuhandaufgaben
			5065 *Erbschaftsangelegenheiten
31	Sonderleistung LOHN	98	Eigenverwaltung (aktuelles Jahr)
	3401 Einrichten einer Lohnbuchführung		9810 Personalwesen
	3440 *Hilfeleistungen beim Lohn		9820 Kanzleirechnungswesen
	3442 Meldungen/Bescheinigungen		9830 *EDV/Organisation
	3444 Lst./Sozialversicherungsprüfungen		9840 interne Weiterbildung
	3431 EDV-Kosten Lohn		9845 Aus- Fortbildung, Schule
	3470 *Sonstiges Lohn		9850 Kanzleiführung
			9860 Sekretariatstätigkeiten
			9880 Ablage/Archiv
35	Abschluß (Bearbeitungsjahr)	99	Fehlzeiten (aktuelles Jahr)
	2410 VA/VermSt (Betrieb)		9920 Krankheit
	2460 Feststellung Einheitswert Grundbesitz		9930 Urlaub
	2470 Sonstige Steuern		9940 Feiertag
	3550 *Abschlußarbeiten inkl. jährlicher betrieblicher Steuererkl.		9970 *Sonstige Fehlzeiten
	3520 *Abschlußvorarbeiten	97	Sonst. EDV - Kosten
	3560 Abschlußbesprechung		9730 EDV allgemein
	3506 Eröffnungsbilanz	60	Wirtschaftsprüfung
	3502 Sonder- / Zwischenbilanz		6000 Wirtschaftsprüfung
	3531 EDV-Kosten Abschluß		

* eine Erfassung mit Notiz ist notwendig!

Quelle: Barz, D. / Wunram, S., 1999, S. 76

In dem Steuerberatungsbüro Becker wird zur Rechnungserstellung das DATEV-Programm PC-KORG verwendet, in welchem die Mitarbeiter alle Tätigkeiten als LEA-Satz mit jeweiliger Dauer erfassen, die sie für einen Mandanten durchgeführt haben.

⁸ vgl. Barz, D. / Wunram, S., 1999, S. 70

Durch einen Vergleich zwischen den durchgeführten Tätigkeiten und dem eigentlichen Auftrag ist es dann möglich, Sonderleistungen, wie das Sortieren und Nachfordern von Belegen für die Monatsbuchführung oder die umfangreiche Klärung offener Fragen durch den Mandanten, zu erkennen und gesondert abzurechnen.

1.3 Zielsetzung und Gang der Arbeit

Ziel dieser Arbeit ist die Beschreibung der Entwicklung eines Systems, welches einerseits für Neumandate, wie auch für Altmandate die Anlage neuer Aufträge und das Ändern bereits bestehender Aufträge leisten kann. Dazu wird zuerst das geplante Pflichtenheft vorgestellt und erläutert. Anschließend werden die notwendigen Datentabellen an einem ER-Modell durch Transformation entwickelt. Danach wird auf die Gestaltung der Benutzerdialoge eingegangen und in diesem Rahmen exemplarisch am Access-Form zur Auswahl der Textbausteine und dem Hauptmenü die dazu notwendigen Arbeitsschritte aufgezeigt. Bei der Beschreibung der Entwicklung des Programmes wird zusätzlich noch ein Augenmerk auf die angewendeten Testverfahren gerichtet.

2 Planung der Entwicklung

2.1 Bisheriger organisatorischer Ablauf in der Kanzlei

Wird ein Programmsystem zur Unterstützung bisheriger betrieblicher Organisation entwickelt, ist es notwendig, in diesem Programm die betrieblichen Organisationsstrukturen der Kanzlei möglichst genau widerzuspiegeln. „Entsprechend der Bedeutung der Ressource Informatik für einen Betrieb muß das IS-Management eine Informationsinfrastruktur bereitstellen, die dazu beiträgt, die Erfolgspotentiale des Betriebes zu sichern und weiter auszubauen“⁹. Ferner kann die Software dabei unterstützen, Daten analytisch auszuwerten, um z.B. Methoden des Controllings anwenden zu können.

In diesem speziellen Fall soll die Informatik helfen, die dem Mandanten versprochene Leistung und die tatsächlich erbrachte Leistung abzustimmen, um eine für den Mandanten

⁹ Hansen, H. R., 1998, S. 115

und für den Mitarbeiter transparentere Rechnungslegung zu garantieren. Um dieses effektiv leisten zu können, muß das zu erstellende Programm unterstützend in den bisherigen betrieblichen Ablauf zu integrieren sein, was es wiederum erforderlich macht, diesen genau zu kennen und vor allem die Schwachstellen aufzudecken. Dazu wird im folgenden der derzeitige Ablauf zur Neumandats- und Auftragserfassung in Augenschein genommen und vor allem die bisher genutzte Software erläutert.

2.1.1 Die Neumandatsanlage

Ein Neumandat beginnt mit einem persönlichen Gespräch, in welchem der Leistungskatalog der Kanzlei in groben Zügen vorgestellt wird, um dann die speziell für diesen Mandanten auszuführenden Tätigkeiten festzuhalten. Des weiteren werden auf einem vorgedruckten Laufzettel handschriftlich sämtliche von den Programmen der DATEV benötigten Daten wie Anschrift, Telefonnummern, Betriebsanschrift, Firmierung, etc., erfaßt. Der Laufzettel wird benötigt, um die Daten nach dem Gespräch in den Programmen ZMSD und PC-KORG einzugeben, wobei die ZMSD dann auch die endgültige Mandantenummer vergibt, die wiederum benötigt wird, um für die hauseigene Buchführung ein Debitorenkonto anzulegen.

Eine Online-Erfassung der Daten mit der bisher verwendeten Software während des Gespräches wird als nicht praktikabel angesehen, da die Programme sehr aufwendig und komplex gestaltet sind und einen hohen Zeitaufwand bei der Erfassung erfordern, welchen man dem Mandanten nicht zumuten mag.

Nicht selten wird bei dieser Prozedur jedoch einiges vergessen, was dann hinterher von dem Sachbearbeiter per Telefonat nachgefragt werden muß. Das bereitet vor allem dann Probleme, wenn zudem noch eine Unterschrift des Mandanten, z.B. für einen Abbuchungsauftrag der monatlichen Gebühr, fehlt.

2.1.2 Die Auftragserfassung und -änderung

Wie sich in einem persönlichen Gespräch mit Herrn Winfried Becker herausstellte¹⁰, wurde bisher keine Auftragserfassung getätigt. Der Mandant hat seine Wünsche verbal formuliert und damit den Auftrag zum Erbringen der gewünschten Leistungen gegeben. Eine Überprüfung kann damit nicht stattfinden, was vor allem bei abweichenden Vorstellungen über den Auftrag zwischen dem Mandant und dem Sachbearbeiter in der Kanzlei zu schwer zu beseitigenden Problemen führen kann; vor allem, wenn Uneinigkeit über die abzurechnenden Leistungen besteht. Die Mandanten interpretieren nicht selten Sonderleistungen als zum Auftrag gehörende Standardleistungen und sind somit nicht gewillt, diese extra zu vergüten.

2.2 Das Pflichtenheft

Aufgabe ist die Entwicklung eines Auftragserfassungssystems (AES), in welchem die Aufträge der Mandate verwaltet werden und jederzeit abrufbar sind. Speziell für Neumandate muß es möglich sein, diese vorerst ohne Mandantenummer zu erfassen, da diese nur aus der ZMSD heraus vergeben werden kann. Daher ist es nötig, für eine einfache Nacherfassung der Mandantenummern zu sorgen. Zusätzlich ist ein vom Benutzer auszulösender automatisch ablaufender Datenabgleich zwischen dem Datenbestand der ZMSD und dem Bestand des zu erstellenden AES geplant.

Des Weiteren soll das System dem Berater helfen, bei einem Neumandat während des ersten persönlichen Gesprächs online alle Punkte einer Mandatsanlage abzulaufen und dabei abzustimmen, welche Arbeiten von ihm ausgeführt werden sollen. Das Programm soll hierzu selbständig alle notwendigen Daten vom Berater in Form eines Assistenten erfragen. Ein wichtiger Punkt ist u.a. die Klärung, ob der Mandant am Lastschriftverfahren teilnehmen möchte, um dann ggf. einen vom Mandanten zu unterschreibenden Formdruck auszuwerfen.

Da ein Mandant Aufträge aus verschiedenen Auftragsarten, wie Lohnbuchhaltung oder Finanzbuchhaltung, erteilen kann, sollen diese in der Auftragserfassung ebenfalls getrennt erfaßbar sein und zusätzlich die Auftragsart in der Auftragsnummer geschlüsselt werden.

¹⁰ s. Anhang 1

Zu jeder Auftragsart soll es eine Auswahl an Textbausteinen geben, welche die auszuführenden Tätigkeiten umschreiben. Zusätzlich soll es möglich sein, zu einer Auftragsart individuelle und möglicherweise nur für dieses eine Mandat gültige Textbausteine zu erstellen. Jeder Textbaustein soll einen Preisschlüssel enthalten, damit es möglich ist, den gesamten Preis eines Auftrages zu ermitteln. Dazu ist es, basierend auf der StBGebV, notwendig, den Gegenstandswert und den zugehörigen Gebührensatz zu speichern¹¹. Die Preisermittlung soll nicht dem internen Controlling der Kanzlei dienen, indem kalkulierter und tatsächlich veranschlagter Preis verglichen werden, sondern dem Mandanten und auch dem Berater nur einen ersten Anhaltspunkt geben.

Ferner müssen Auswertungen erstellt werden, welche die erfaßten Daten in übersichtlicher Form darstellen, so daß die gespeicherten Daten einfach von den Mitarbeitern der Kanzlei gepflegt werden können.

2.3 Wahl der Programmierumgebung

Da umfangreiche Datenmengen zu verwalten sind, auf welche zudem zahlreiche Anfragen laufen sollen, ist ein DBMS unumgänglich. In der Kanzlei ist bereits das Office-Paket 97 der Firma Microsoft in der „Professional Version“ installiert, wodurch das Programm Access auf jedem Rechner verfügbar ist und sich somit als Entwicklungsumgebung anbietet. Vorteilhaft bei diesem Programm ist die Möglichkeit, Forms zu erstellen, die einen von der Datenbank getrennten Benutzerdialog erlauben. Der Programmierer kann diese Forms selbst gestalten und mit Buttons, Drop-Down-Listen und ähnlichem versehen, so daß eine Bedienung des AESs von den Benutzern weder Kenntnisse in der Verwaltung von Daten noch in der Bedienung eines DBMS voraussetzt. Zudem kann der Programmierer in der Sprache Visual Basic (VB) eigene kleinere Applikationen in das Programm einbinden, womit eine große Flexibilität gewährleistet ist. Ferner ist es möglich, Daten von Access ohne Definition von Exportschnittstellen in andere Office-Anwendungen zu exportieren.

¹¹ vgl. StBGebV, 1998, Beispielsweise §35, S. 84f. (s.a. Anhang 2)

2.4 *Netzwerkfähigkeit und -einsatz des AES*

Durch den fortschreitenden Einsatz von Netzwerken in Büros ist mittlerweile die Netzwerkfähigkeit von Software eine Grundvoraussetzung für deren Einsatzfähigkeit geworden. Die Vernetzung aller im Büro eingesetzter PCs, ob mit einem extra File-Server, oder als Peer-to-Peer-Netz, erfreut sich großer Beliebtheit, da nicht nur die Netzwerkkarten und das restliche nötige Equipment erschwinglich geworden sind, sondern die Vernetzung an sich viele weitere Vorteile bietet, wenn man allein die Einsatzmöglichkeiten eines internen Mailingsystems betrachtet.

Oft macht selbst die Art der Datenhaltung eine Vernetzung aller PCs zwingend erforderlich. Die betrieblichen Daten, seien es Kunden- oder Auftragsdaten, die Rechnungserstellung und Lagerverwaltung, etc., können nur durch den Einsatz eines Netzwerkes allen Mitarbeitern zur Verfügung stehen und trotzdem zentral verwaltet werden.

Unter diesen Gesichtspunkten ist auch bei der Erstellung des AES darauf zu achten, daß ein netzwerkfähiger Einsatz der Software möglich ist. Die Verwendung von Access als DBMS schafft hierbei eine grundlegende Voraussetzung, da Access alle nötigen Algorithmen des Netzwerkmanagements beherrscht. Hierzu gehört vor allem das Sperren von Datensätzen, die gerade bearbeitet werden und die Möglichkeit, die implementierte Software vor unbefugten Zugriffen zu schützen.

Durch die Vernetzung werden oft auch unbewußt Möglichkeiten geschaffen, Außenstehenden einen unbemerkten Zugang zum Datenbestand des Betriebes zu schaffen. Insbesondere in einer Steuerkanzlei, wo zum Teil äußerst sensible Daten gespeichert werden, ist ein konsequent betriebener Datenschutz notwendig. Als Mehrbenutzersystem wäre somit eine Anmeldung nach dem Start jeglicher Software notwendig, um die Zugriffsrechte des Mitarbeiters zu prüfen. Davon wird bei dem AES kein Gebrauch gemacht, da sämtliche im AES implementierte Funktionen allen Mitarbeitern uneingeschränkt zur Verfügung stehen sollen. Zudem kann das AES von den Sicherheitsvorkehrungen für den Datenbestand der übrigen Software partizipieren. So muß sich jeder Mitarbeiter bereits beim Start des PCs mit einer expliziten Netzanmeldung für die nachträgliche benutzerabhängige Zuteilung von Netzressourcen authentifizieren.

3 Datenbestand

3.1 *Benötigte Daten*

Bevor eine Tabelle in Access entstehen kann, ist es notwendig, sich genaue Gedanken darüber zu machen, welche Daten benötigt werden, und in welcher Beziehung diese zueinander stehen. Es ist klar, daß auf jeden Fall bestimmte Daten zu Mandanten gespeichert werden müssen. Dabei muß vor allem die Postanschrift gespeichert sein, da im Ergebnis Aufträge gedruckt werden sollen, welche die genaue Adresse enthalten müssen. Da zudem das AES auch Formulare zum Bankeinzug ausgeben soll, müssen ebenso Bankdaten gespeichert werden. Aus diesen Überlegungen ergibt sich daher für den Entity-Typ „**Mandantendaten**“ mindestens die folgende Aufstellung an Attributen:

- Mandantenummer
- Name (Vorname, Nachname)
- Firmenname (Komplett mit Rechtsform)
- Adresse (Straße, Hausnummer, PLZ, Ort)
- Bankverbindung (BLZ, Bankname und Kontonummer),
falls Bankeinzug gewünscht.

Ferner soll das AES den Leistungskatalog der Kanzlei beinhalten, der je nach Kundenwunsch teilweise in dessen Auftrag übernommen wird. Zu diesem Zweck wird der Leistungskatalog in Form von Textbausteinen in das AES aufgenommen, die zudem einen Preis enthalten müssen.

Die Preiskalkulation ist dabei nicht so direkt fest zu machen, wie bei anderen Dienstleistern, da die steuerberatenden Berufe an die StBGebV gebunden sind. In dieser Verordnung sind die als Textbausteine verwendbaren abrechenbaren Tätigkeiten eines Steuerberaters aufgelistet und mit einem Verrechnungsschlüssel versehen. Dieser besteht aus einem Gegenstandswert, einer Zuordnungstabelle und einem zugehörigen Faktor. Beispielsweise kann als Gegenstandswert der Jahresumsatz des Mandanten herangezogen werden, aus welchem dann mit der Hilfe eines Tabellenwerkes die maximale volle Monatsgebühr ermittelt wird. Von der vollen Gebühr wird dann mit der Hilfe eines vom Steuerberater festgelegten Faktors, dessen Spanne ebenfalls durch die StBGebV festgelegt ist, die tatsächli

che Monatsgebühr ermittelt. Daher ergeben sich für die „**Textbausteine**“ folgende zu speichernde Elemente:

- Text der Tätigkeit
- Auftragsart
- Gegenstandswert
- Basierende Tabelle
- Gebührensatz / Zeitgebühr

Des Weiteren sind noch die Aufträge zu erfassen. Grundsätzlich wird ein Auftrag immer genau einer Auftragsart, wie Lohnbuchhaltung oder Jahresabschluß, zugeordnet, und enthält daher nur Textbausteine aus der entsprechenden Auftragsart. Zusätzlich muß noch der zugehörige Mandant und eine fortlaufende Auftragsnummer gespeichert werden. Daraus ergibt sich dann für den Entity-Typen „**Aufträge**“ folgende Zusammenstellung:

- Auftragsnummer
- Auftragsart
- Mandant
- Textbausteine des Auftrages

3.2 Datenherkunft

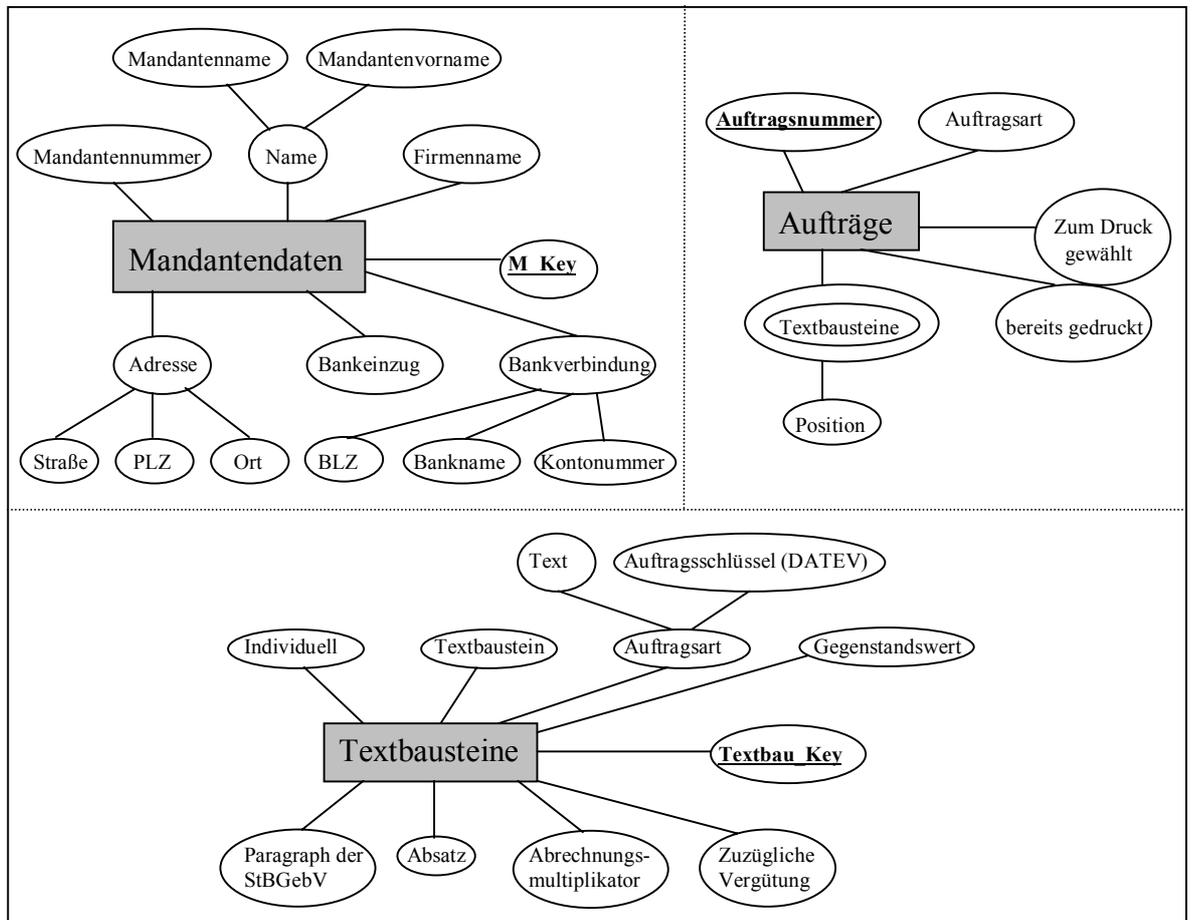
Die Mandantendaten können zur Erstbestückung des Programms aus der ZMSD bezogen werden. Dabei ist zu beachten, daß die Mandanten, die am Lastschriftverfahren teilnehmen, nur aus PC-KORG zu ermitteln sind, da eine entsprechende Markierung in der ZMSD nicht vorhanden ist. Die Textbausteine dagegen müssen manuell auf der Grundlage der StBGebV und des Leistungskatalogs der Kanzlei eingepflegt werden.

Ob bereits bestehende Aufträge jemals nacherfaßt werden, bleibt fraglich, da bisher weder konsequent Aufträge schriftlich entgegengenommen, noch eine elektronische Speicherung durchgeführt wurde. Die manuelle Eingabe dürfte damit sehr zeitaufwendig und zudem kostenintensiv sein.

3.3 ER-Modell der Daten

Nach den grundlegenden Überlegungen zu den zu speichernden Daten folgt nun im nächsten Schritt die Erstellung eines Entity-Relationship-Modells. Dabei wird aus Gründen der Übersicht zunächst nur auf die Entity-Typen und ihre Attribute¹² eingegangen und dann auf die Relationships¹³ zwischen den Entity-Typen.

Abbildung 2: Die Entity-Typen mit ihren Attributen



Quelle: eigene Darstellung, in Anlehnung an DUDEN Informatik, 1993, S.231ff.

In diesem Modell sind einige Attribute hinzugefügt worden, die nicht durch das Pflichtenheft festgelegt sind, sondern durch die zu implementierenden Funktionen notwendig werden. So handelt es sich bei den Attributen „bereits gedruckt“ und „zum Druck gewählt“

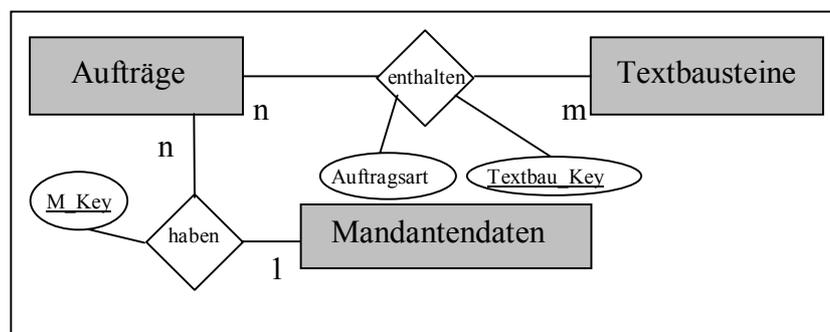
¹² s. Abbildung 2

¹³ s. Abbildung 3

um boolesche Attribute, die nur die Werte „Ja“ und „Nein“ annehmen können. Sie werden eingeführt, um bei der Druckaufbereitung bereits gedruckte Aufträge auszublenden, und dem Benutzer des weiteren die Möglichkeit zu bieten, nur bestimmte Aufträge zum Drucken auswählen zu können.

Ebenso ist das Attribut „Bankeinzug“ des Entity-Typs „Mandantendaten“ vom Datentyp Bool, da das Attribut „Bankverbindung“ nur dann gültige Werte enthält, wenn der Mandant am Lastschriftverfahren teilnimmt.

Abbildung 3: Die Relationships



Quelle: eigene Darstellung, in Anlehnung an DUDEN Informatik, 1993, S.231ff.

Der Entity-Typ „Mandantendaten“ hätte als Schlüsselement die Mandantenummer erhalten können, welche in der Kanzlei einzigartig ist. Da aber Mandate vor allem bei einer Neuanlage auch ohne eine Mandantenummer erfassbar sein müssen, ist dieses Attribut in dem AES nicht eindeutig und somit als Schlüssel nicht brauchbar. Aus diesem Grund mußte für diesen Entity-Typ ein eigener Schlüssel „M_Key“ eingeführt werden.

Das zum mehrwertigen Attribut „Textbausteine“ gehörende Attribut „Position“ des Entity-Typen „Aufträge“ nimmt die Textposition des Bausteines innerhalb des Auftrages auf, um es dem Benutzer zu ermöglichen, auf die Reihenfolge der Textbausteine im Dokument Einfluß nehmen zu können. Bei diesem Entity-Typen kann die Auftragsnummer als Schlüssel dienen, da diese eindeutig ist.

Der Entity-Typ „Textbausteine“ enthält als Attribut auch den zugehörigen Paragraphen der StBGebV, wenn der Textbaustein daraus entnommen worden ist. Andernfalls ist dieses Attribut Null. Zusätzlich können individuell erfaßte Textbausteine durch das boolesche Attribut „individuell“ gekennzeichnet werden. Ferner ist hier das Schlüsselattribut „Textbau_Key“ einzuführen, da der Entity-Typ „Textbausteine“ ansonsten kein eindeutiges Attribut enthält.

Die Beziehungen der Entity-Typen zueinander sind in Abbildung 3 dargestellt. Es fällt auf, daß zu dem Beziehungstyp „Aufträge enthalten Textbausteine“ noch das Attribut „Auftragsart“ hinzugefügt wurde, da ein Auftrag nur Textbausteine aus seiner jeweiligen Auftragsart enthalten soll.

3.4 Transformationsprozeß und Normalisierung

Im nächsten Schritt zur Erstellung des AES müssen nun aus dem ERM durch Transformation in das Relationenmodell die für Access brauchbaren Tabellen erzeugt werden. Danach ist es sinnvoll, die Relationen zu normalisieren, um Redundanzen zu vermeiden und um Inkonsistenzen vorzubeugen¹⁴. „Es existieren Regeln, um Daten zu strukturieren, bevor sie dann in eine Datenbank überführt werden“¹⁵. Dieser als Normalisierung bezeichnete Prozeß sieht vor, daß nach der Überführung des ERM in das Relationenmodell jede Relation auf den Grad seiner Normalisierung hin überprüft und somit sukzessive auf die dritte Normalform gebracht wird. Die vierte und fünfte Normalform finden in der Praxis selten Anwendung, da sie recht kompliziert sind.

Die Überführung des ERM in das Relationenmodell ist in Abbildung 4 dargestellt. Für die Darstellung wurde bereits die Normalisierung durchgeführt. Da keine Beispieldaten existieren, um die Relationen zu füllen, und diese auch nicht zwingend notwendig sind, um die Regeln der Normalisierung anzuwenden, wird auf die Darstellung von Daten im Relationenmodell verzichtet und nur die Tabellenform vorgestellt.

Im Relationenmodell wird deutlich, wie die Relationships die Entity-Typen beeinflussen, da an die Stelle einiger Attribute nun Schlüsselwerte treten. Zudem begünstigt die entstandene Tabellenform den iterativen Implementierungsprozeß neuer, bisher nicht geplanter Funktionen.

Im folgenden wird nun erläutert, wie der Normalisierungsprozeß zu dem in Abbildung 4 dargestellten Relationenmodell führt. Dabei wird nicht, wie sonst üblich, ein erster Entwurf des Relationenmodells für den Normalisierungsprozeß verwendet, sondern direkt das ERM. Dennoch wird der Begriff der Relation im folgenden maßgeblich sein.

¹⁴ vgl. Hansen, H. R., 1998, S. 949 und DUDEN Informatik, 1993, S. 161

¹⁵ Becker, J., 1997, S.25

Um die Normen der ersten Normalform zu erfüllen, müssen Wiederholungsgruppen eliminiert und in eine eigene Relation überführt werden¹⁶. In Abbildung 2 ist zu erkennen, daß das mehrwertige Attribut „Textbausteine“ des Entity-Typen „Aufträge“ eine Wiederholungsgruppe darstellt, da es innerhalb eines Datensatzes von „Aufträge“ mehrfach vorkommt und damit nicht voll funktional abhängig vom Primärschlüssel ist¹⁷. Daher muß das mehrwertige Attribut „Textbausteine“ in eine eigene Relation überführt werden. Als Schlüssel der neuen Relation wird die Auftragsnummer herangezogen, wie in Abbildung 4 dargestellt. Zudem wird die Überführung eines mehrwertigen Attributes in eine eigene Relation bereits von den Regeln des Transformationprozesses gefordert.

Die zweite Normalform wird erreicht, wenn alle Nichtschlüsselattribute vom gesamten Primärschlüssel voll funktional abhängig sind¹⁸. Das bedeutet, daß alle Nichtschlüsselattribute in eine eigene Relation überführt werden, wenn sie nur von einem Teil des Primärschlüssels abhängig sind. Ferner gilt die Regel: „Wenn eine Relation in 1. Normalform ist und nur einen Schlüssel hat, ist sie automatisch in der 2. Normalform“¹⁹. Dies trifft auf alle Relationen des AES zu. Somit ist auch die zweite Normalform erreicht.

Damit eine Relation auch der dritten Normalform entspricht, darf kein Nichtschlüsselattribut transitiv von einem Schlüsselattribut abhängen²⁰. Dies ist jedoch z.B. für das Attribut „Bankname“ im Entity-Typ „Mandantendaten“ der Fall. Das identifizierende Merkmal des Banknamens ist nicht der Schlüssel „M_Key“ der Relation „Mandantendaten“, sondern das Attribut „BLZ“, welches für den Namen der Bank steht. Somit müßte der Bankname in eine eigene Relation mit dem Schlüssel „BLZ“ überführt werden, damit die Relation „Mandantendaten“ der dritten Normalform entspricht. In diesem Fall wäre auch ein eigener Dialog zur Pflege der Banknamen mit zugehöriger „BLZ“ zu implementieren.

Dies stellt einen umfangreichen Aufwand dar, der nicht gerechtfertigt erscheint, da zudem der Name der Bank für das AES nicht entscheidend ist und vor allem keine Selektion auf den Namen erfolgen wird. Daher kann man sich diese Redundanz erlauben, um den

¹⁶ vgl. Becker, J., 1997, S. 25f.

¹⁷ vgl. Becker, J., 1997, S. 26

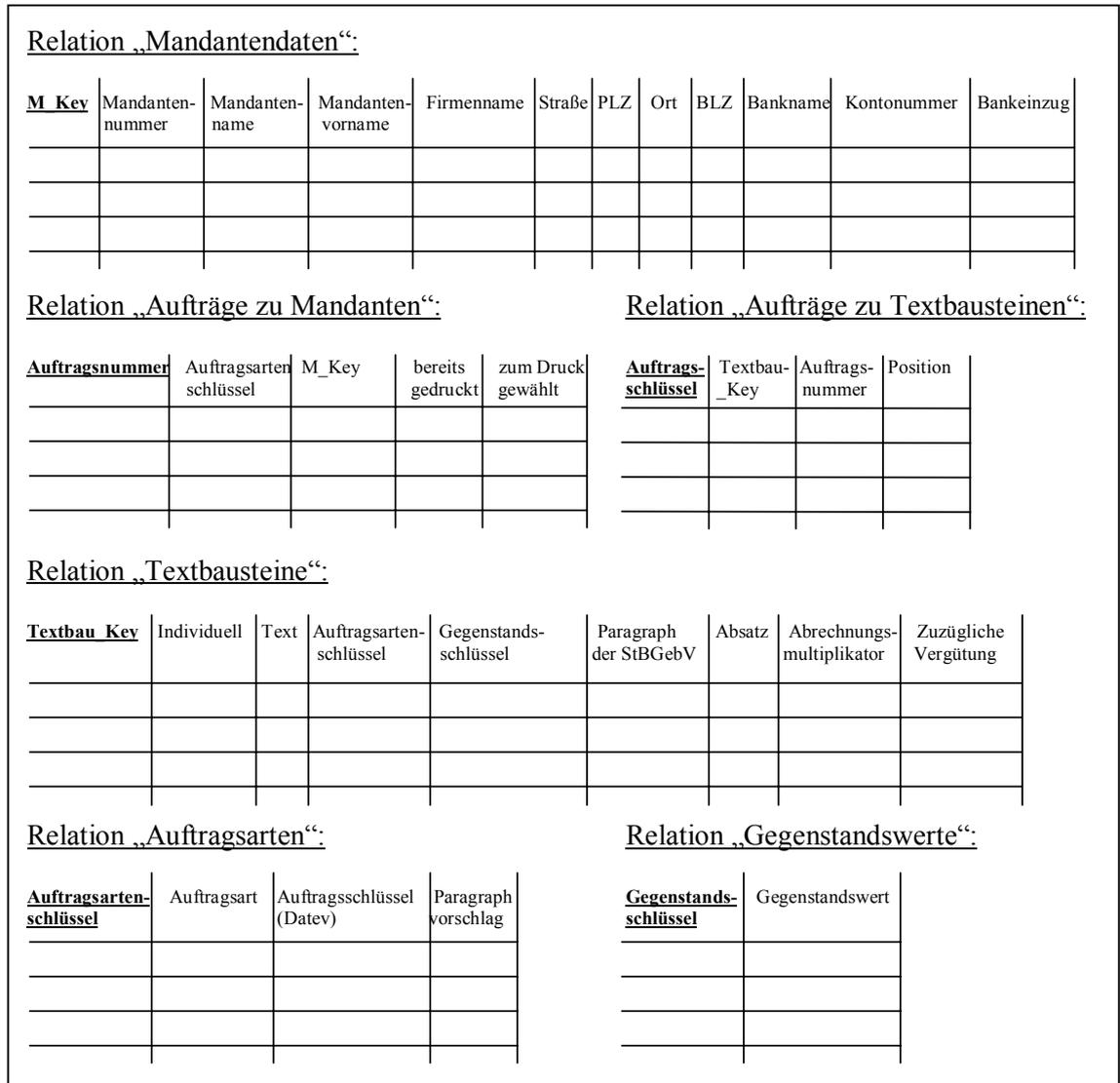
¹⁸ vgl. Becker, J., 1997, S. 27f.

¹⁹ Becker, J., 1997, S. 27

²⁰ vgl. Becker, J., 1997, S. 28f.

Aufwand für die Programmierung und damit auch mögliche Fehlerquellen gering zu halten.

Abbildung 4: Das Relationenmodell des AES

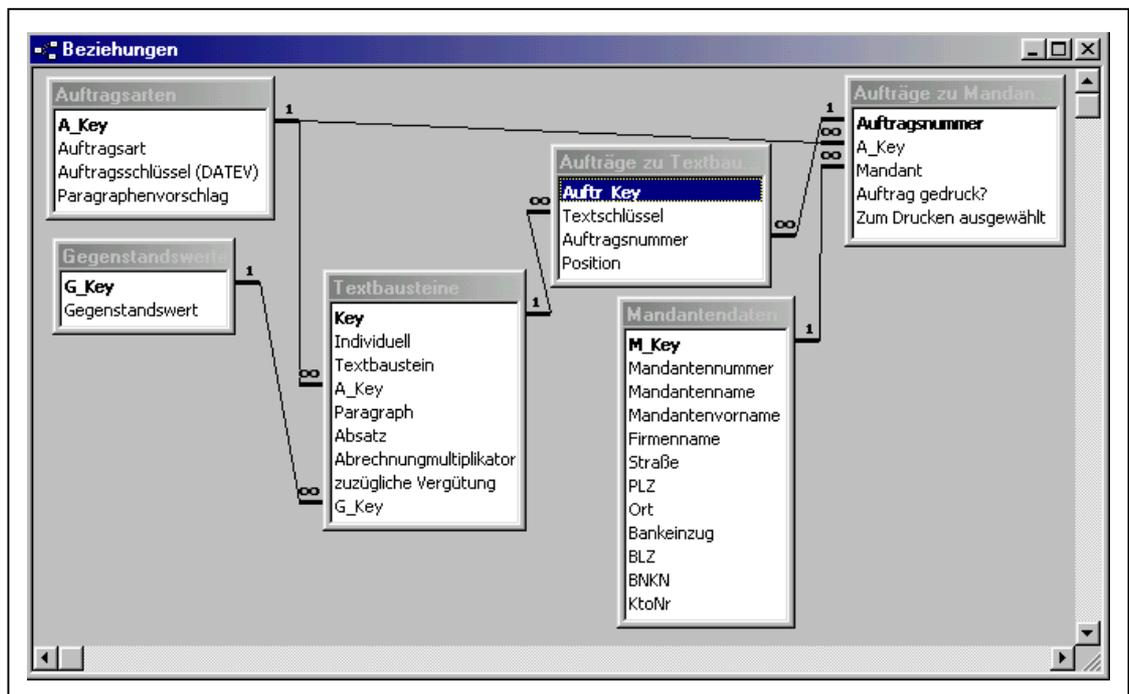


Quelle: eigene Darstellung, in Anlehnung an DUDEN Informatik, 1993, S. 164

Auch die Attribute „Gegenstandswert“ und „Auftragsart“ des Entity-Typen „Textbausteine“ stellen eine Verletzung der dritten Normalform dar, da sie ebenfalls transitiv vom Primärschlüssel abhängig sind. In diesem Fall ist die Überführung in eine eigene Relation sinnvoll. Dazu ist es notwendig, jeweils ein neues Schlüsselattribut einzuführen, da bisher keines existiert. Nach dieser letzten Überführung sind alle Relationen in der dritten Normalform und es ergibt sich das in Abbildung 4 dargestellte Relationenmodell.

In der Relation „Auftragsarten“ wurde ein Paragraphenvorschlag eingeführt, um dem Benutzer bei der Erfassung eines Textbausteines nach Festlegung der Auftragsart den zugehörigen Paragraphen der StBGebV vorzuschlagen. Dieser kann dann geändert werden, sollte der Textbaustein gar nicht oder unter einem anderen Paragraphen in der StBGebV stehen.

Abbildung 5: Darstellung der Beziehungen unter Access



Quelle: Screenshot aus Access 97, Beziehungen im AES

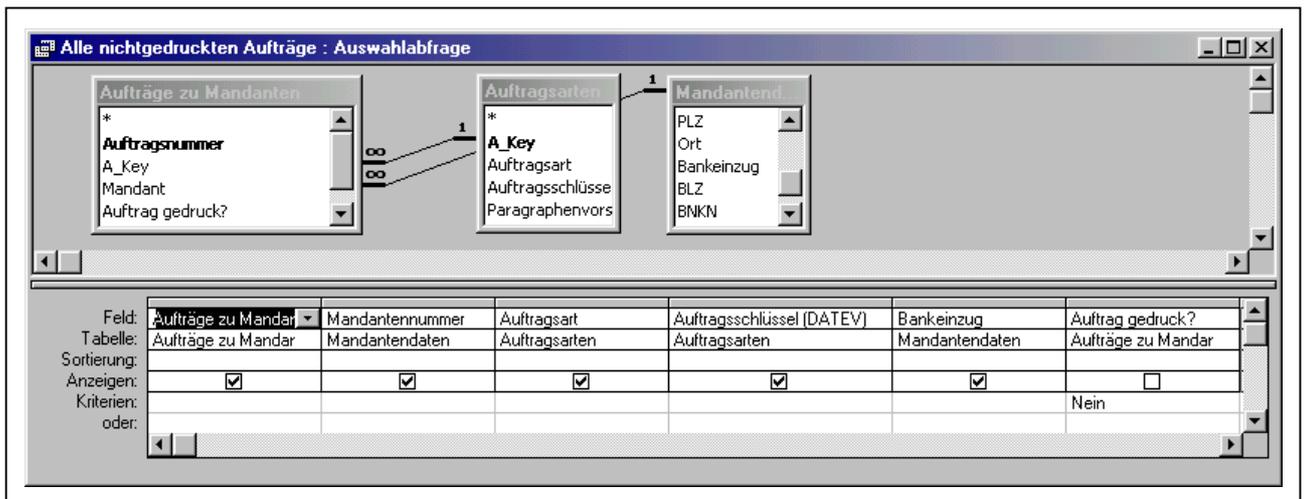
Das Relationenmodell kann nun nach Access übernommen werden. Die Relationships werden unter Access mit der Festlegung von Beziehungen zwischen den Schlüsseln der Tabellen untereinander nachempfunden. Die unter Access dargestellten Beziehungen sind jedoch nicht mit den Relationships des ERM zu verwechseln. Sie dienen dem DBMS lediglich dazu, die Zusammenhänge der Tabellen untereinander festzulegen, und um Integritätsregeln einzuführen. Nach der Übernahme der Tabellen und Festlegung der Beziehungen mit referentieller Integrität²¹ ergibt sich die in Abbildung 5 gezeigte Darstellung. Einige Attributbezeichnungen, vor allem die Schlüsselemente, wurden bei der Übernahme nach Access aus ökonomischen Gründen abgekürzt.

²¹ s. Glossar

3.5 Erstellung von Abfragen unter Access

Das designen von Abfragen stellt eine Hauptaufgabe im AES dar. So ist für jedes Formular und für jeden Bericht eine Abfrage als Grundlage der Datenherkunft zu erstellen.

Abbildung 6: Alle nicht gedruckten Aufträge ermitteln



Quelle: Screenshot aus Access 97, Auswahlabfrage aus dem AES

Dabei wird vornehmlich ausgewählt, welche Datenelemente angezeigt werden sollen. Kommen dabei Elemente in unterschiedlichen Tabellen bzw. Relationen vor, muß eine Verknüpfung erstellt werden, die in erster Linie auf der in Access bereits eingerichteten referentiellen Integrität beruht und durch Auswahlkriterien komplettiert werden kann.

Unter Access wird statt der reinen SQL-Befehle²² eine graphische Oberfläche genutzt, welche die SQL-Befehle visualisiert. Dadurch ist das Erstellen von Abfragen auf sehr intuitive Weise möglich.

3.6 ZMSD-Datenabgleich

Die ZMSD stellt in der Kanzlei den zentralen und wichtigsten Speicherort für Mandantendaten dar. Nur in diesem Programm wird die Mandantennummer vergeben und sämtliche für die DATEV-Software nötigen Stammdaten eines Mandates gepflegt. So sind mehrere

²² s. Glossar

Adressen wie Rechnungsanschrift und Firmenanschrift, mehrere Telefon- und Faxnummern, die Steuernummer und Anschrift des zuständigen Finanzamtes, etc., zu verwalten. Das AES benötigt nur einen Bruchteil dieser Daten. Da jedoch Mandantendaten durch die Einführung des AES doppelt gespeichert werden, ist es nötig, diese regelmäßig mit dem Datenbestand der ZMSD abzustimmen.

Erforderlich ist ein Datenabgleich vor allem nach einer Neumandatsanlage über das AES. Die dort erfaßten Daten müssen in die ZMSD übernommen werden, wo dann eine Mandantenummer vergeben wird, die nachträglich in den Datenbestand des AES eingepflegt werden muß.

Es ist wünschenswert, diese Prozesse so weit wie möglich zu automatisieren. Vor allem im Hinblick auf die Probleme des Medienbruchs²³ sollte der Datenabgleich möglichst auf digitaler Ebene stattfinden. Unter Access ist es durchaus möglich, Daten zu exportieren und auch zu importieren. Empfehlenswert ist dabei die Definition einer ASCII-Exportschnittstelle, die auf dem Prinzip der Komma-Delimited-Format-Datei basiert. Hierbei werden alle Datensätze durch Kommata voneinander getrennt. Durch die festgelegte Anzahl der Elemente eines Datentupels ist es dann möglich, jeden einzelnen Datensatz auszulesen und in das AES zu übernehmen, und umgekehrt. Da Access gute Verbindungen zu den anderen Office-Produkten unterhält, ist ein Export zu MS-Excel ebenso vorteilhaft, da nicht selten Programme gerade aus Excel Daten importieren können.

Da jedoch bisher keine Möglichkeit bekannt ist, wie der Datenbestand der ZMSD von einer anderen Software beeinflußt werden kann, bietet das AES nur das Formblatt „ZMSD-Neuanlage“, welches ausgedruckt und an den zuständigen Mitarbeiter gereicht werden kann. Der somit entstandene Medienbruch muß vorerst in Kauf genommen werden.

Der Datenexport dagegen ist über das ZMSD-Auskunftssystem möglich, welches selbst auf Access basiert. Der Programmierer erstellt hierzu einen Bericht mit individuellen Feldern, der auf einem bereits erzeugten Standardbericht der DATEV basiert. Es werden jedoch nicht alle in der ZMSD verwendeten Datenfelder bereitgestellt, sondern nur eine Auswahl derer, die bei der Erstellung des Standardberichts nötig waren²⁴. Diese Abfrage kann nicht verändert werden, wodurch neue Felder nicht hinzugefügt werden können. Die

²³ s. Glossar

²⁴ s.a. Kapitel 4.3, insbesondere die Erläuterungen zum Erstellen eines Berichtes unter Access

ses Problem läßt sich aber dadurch lösen, daß mehrere Exporte basierend auf unterschiedlichen Berichten durchgeführt werden, bis alle benötigten Daten exportiert worden sind.

Bei der Erstellung des Berichts muß der Programmierer keine Zeit auf das Aussehen verwenden. Es genügt, wenn alle Felder, die exportiert werden sollen, im Datenbereich des Access-Berichtes in richtiger Größe zu finden sind, wobei die Position nicht entscheidend ist. Danach kann der Bericht mit Daten gefüllt und z.B. nach Excel exportiert werden. Dabei werden die Namen und die Größen der Felder im Bericht als Beschriftung und Breite der Felder in das Excel-Formular übernommen.

Es kann nun im AES eine Funktion implementiert werden, die aus den in Namen und Datenfeldern festgelegten und vom ZMSD-Auskunftssystem erzeugten Excel-Dokumenten sämtliche Datensätze stur importiert. Sinnvoll wäre eine solche Funktion jedoch nur zur Erstbestückung. Daher ist sie so zu erweitern, daß sie bereits vorhandene Datensätze überschreiben kann. Dadurch sind jedoch auch entsprechende Fehlerüberprüfungen zu implementieren, damit nicht versehentlich bereits manuell korrigierte Datensätze überschrieben werden.

Denkbar ist es auch, die Funktion als Überwachungsfunktion zu nutzen. Dabei soll während des Importes kein Datensatz verändert, wohl aber ungleiche oder fehlende Datensätze in einer Liste aufgeführt werden. Damit wäre es möglich, den Datenbestand des AES mit dem der ZMSD zu vergleichen und Unstimmigkeiten entweder im AES nach Bestätigung einer entsprechenden Abfrage des Programmes sofort, oder manuell in der ZMSD später zu korrigieren.

Die Implementierung einer solchen Funktionalität ist bisher jedoch nur in Planung. Es ist noch genau zu prüfen, wie der Benutzerdialog und die Funktion selbst zu gestalten sind, damit die Mitarbeiter der Kanzlei sicher damit umgehen können. Vor allem ist im Vorfeld zu klären, in welcher Form das Ergebnis des Datenabgleichs präsentiert werden muß, damit die nachträgliche Pflege ungleicher Datensätze zügig und sicher vorgenommen werden kann.

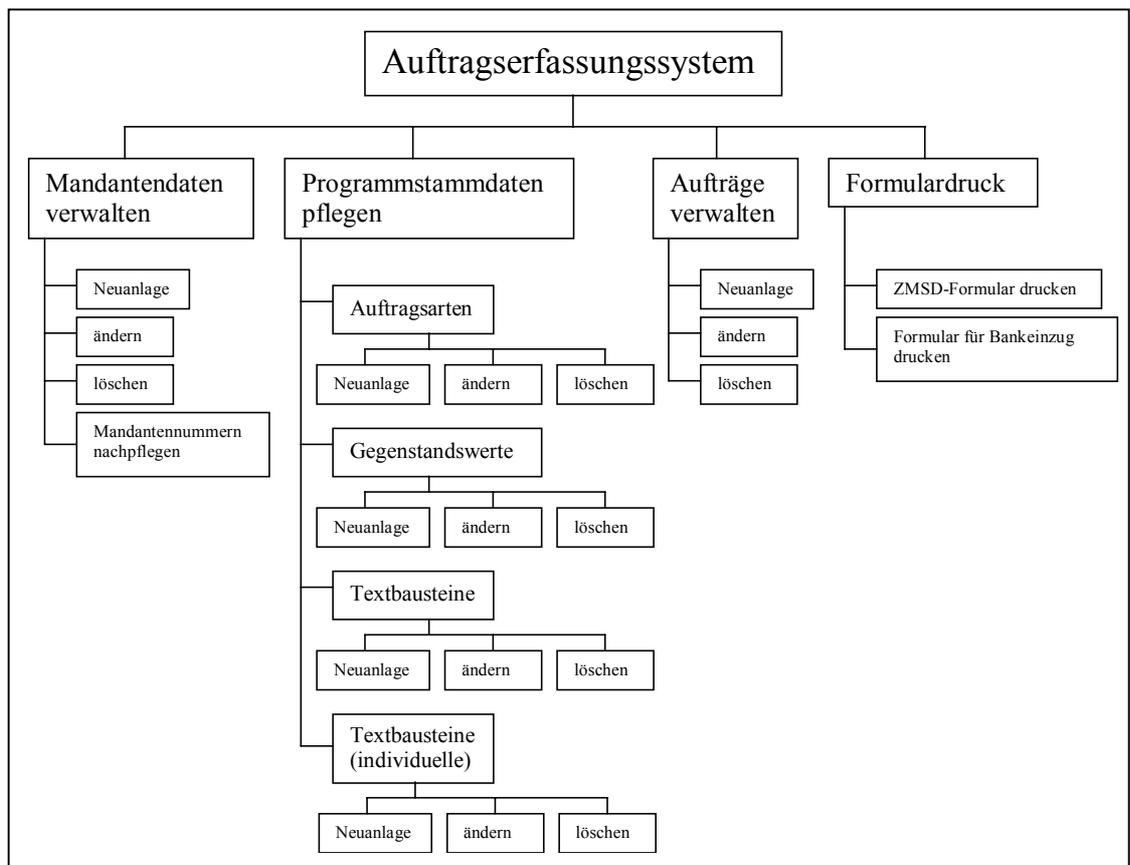
Grundsätzlich gilt, daß bei einem Datenabgleich auf gar keinen Fall Inkonsistenzen innerhalb der Daten des AES entstehen dürfen. Ferner sind die Regeln der unter Access eingeführten referentiellen Integrität zu beachten.

4 Benutzerdialoge

4.1 Hierarchische Funktionsgliederung des Programmes

Auf der Basis des Pflichtenheftes, der erstellten Relationen und den vorangegangenen Überlegungen kann das Programm nun in einzelne Elemente gegliedert werden.

Abbildung 7: Hierarchische Funktionsgliederung des AES



Quelle: eigene Darstellung, in Anlehnung an DUDEN Informatik, 1993, S. 655f., „hierarchische Modularisierung“

Dabei besteht ein Großteil der Gliederungsstufen aus den zu implementierenden Benutzerschnittstellen zur Verwaltung der Datenbestände, welche unter Access über einen Benutzerdialog gesteuert werden können. Diese als „Forms“ oder „Formulare“ benannten Objekte basieren auf der für graphische Systeme üblichen Fenstertechnik. Beispielsweise sind Formulare zur Verwaltung der Mandantendaten sowie der Auftragsverwaltung einzuplanen.

Des weiteren sind Funktionen zur Programmdatenpflege und zum Drucken erfaßter Daten zu implementieren. In Abbildung 7 sind alle notwendigen Programmelemente in einer hierarchischen Funktionsgliederung zusammengestellt.

4.2 Gestaltung der Forms

Bei der Erstellung der Benutzeroberfläche ist darauf zu achten, daß sie einheitlich und von dem Benutzer durchschaubar ist. Dies wird durch Vereinheitlichung der Anordnung von Buttons und deren Icons, sowie durch das Einhalten von bisher durch andere Applikationen geschaffenen Standards gewährleistet.

Die möglichen Eingabegeräte, Maus und Tastatur, müssen bei der Gestaltung der Benutzerdialoge auf ihre Eignung und vor allem ihren möglichen und wahrscheinlichen Einsatz hin überprüft werden. Da sich bei graphischen Oberflächen vor allem die Maus als Eingabegerät anbietet, um sich durch das Programm zu bewegen, sollte diese einheitlich Unterstützung finden. Die Tastatur dagegen sollte nur dann zum Einsatz kommen, wenn der Benutzer Daten erfassen möchte.

Nichtsdestoweniger muß man hierbei auf die Ergonomie achten und die Eingabegewohnheiten der Benutzer berücksichtigen. So sollte es weiterhin möglich sein, die verschiedenen Eingabefelder durch Drücken der Tabulatortaste zu erreichen. Wäre ein Wechseln der Eingabefelder nur durch die Maus möglich, ist der Anwender gezwungen, seine Hände wieder von der Tastatur zu nehmen. Um ein flüssiges und zügiges Erfassen von Daten zu ermöglichen, sollte z.B. bei dem Abschluß eines kompletten Datensatzes das Aufrufen eines neuen, leeren Satzes automatisch ablaufen, bei Änderung eines Datensatzes dagegen nicht.

Unter Access gibt es des weiteren die Möglichkeit, für jegliche in Forms verwendete Objekte einen „Tip-Text“ zu erstellen, der immer dann erscheint, wenn der Anwender den Mauszeiger auf einem Objekt ruhen läßt. Davon sollte bei der Erstellung von Formularen reichlich Gebrauch gemacht werden. Die Tip-Texte gestatten es dem Anwender, sich über die Funktion eines Objektes zu informieren. Dadurch können Dialoge selbsterklärend gestaltet werden, was eine umfangreiche Schulung der Anwender auf das Notwendigste reduzieren kann. Vor allem im Hinblick auf die durch Schulungen verursachten Kosten ist

eine anwenderfreundliche, zweckmäßige und selbsterklärend programmierte Oberfläche sinnvoll.

Bei der Erstellung von Forms ist auch im Vorfeld zu prüfen, ob an alle notwendigen Elemente gedacht wurde und wie diese angeordnet werden, da Platz in Formularen oft nur begrenzt zur Verfügung steht. Ebenso ist die vornehmlich verwendete Bildschirmauflösung zu berücksichtigen, damit das Formular komplett sichtbar ist.

Um zudem den programmiertechnischen Aufwand minimal zu halten, ist auch zu prüfen, ob Teile eines Formulars mehrfach Verwendung finden können und somit die Einführung von Unterformularen sinnvoll ist. Diese brauchen nur einmal erstellt zu werden und können dann über einen entsprechenden Platzhalter im Hauptformular zur Anzeige gebracht werden.

Da ein Formular unter Access immer nur auf einer Abfrage bzw. einer Tabelle basieren kann, ist die Einführung von Unterformularen sogar zwingend erforderlich, wenn aufgrund einer Auswahl im Hauptformular weitere, nicht zum aktuellen Datensatz gehörende Datensätze mit angezeigt werden sollen. Z.B. müssen im Formular zur Auftragserfassung²⁵ die Textbausteine in einem Unterformular angezeigt werden, da sie erst durch die im Hauptformular ausgewählte Auftragsart festgelegt werden.

4.2.1 Probleme des Multithreadings unter Access

Öffnet Access ein Form, ermittelt es die in diesem Form darzustellenden Datensätze und speichert sie zwischen. Werden nun durch das gleichzeitige Öffnen mehrerer Forms Datensätze verändert, werden diese Änderungen nicht an andere bereits geöffnete Forms weitergeleitet. Der Programmierer muß explizit mit dem VB-Befehl „Requery“ das Form erneut zum Einlesen der Datensätze zwingen.

Dies macht jedoch nur dann Sinn, wenn die Programmlogik auch wissen kann, daß sich Datensätze verändert haben. Daher muß der Aufruf von Formularen von dem Programm überwacht werden, so daß der Benutzer nicht selbst mehrere Forms öffnet und damit an der Programmlogik „vorbei“ Datensätze verändert.

²⁵ s.a. Kapitel 4.2.2 und Abbildung 8

Um selbst die Kontrolle über das Öffnen und Schließen von Forms zu haben, wurden die Fensterschaltflächen „MinMax“²⁶ und „Fenster schließen“²⁷ deaktiviert und statt dessen ein eigener Button zum Schließen von Forms eingeführt, welcher immer ganz rechts in der Kopfzeile des Forms zu finden ist.

Des weiteren muß ein Hauptmenü eingeführt werden. Damit wird nicht nur der Benutzerfreundlichkeit Rechnung getragen, sondern auch die Möglichkeit geschaffen, die Aufrufe von Formularen programmtechnisch zu überwachen. Startet Access das AES, ruft es ein als Startformular festgelegtes Formular auf, welches in diesem Fall „Menü_Start“ heißt. Dieses Form ist nicht an eine Tabelle gebunden und stellt damit lediglich einen Benutzerdialog dar, ähnlich einem Fenster. Der Startdialog enthält neben einer Copyright-Meldung nur Buttons, die durch Anklicken weitere Forms öffnen. Da es möglich ist, mehrere Forms gleichzeitig zu öffnen, was u.U. wie oben angeführt fatale Folgen haben könnte, muß man hier bei der Programmierung besondere Vorsicht walten lassen. Die einfachste Möglichkeit, das mehrfache Aufrufen von Formularen zu unterbinden, ist die stets auf dem Bildschirm zentrierte Ausgabe. Damit wird das Startmenü verdeckt und ist nicht mehr zugänglich.²⁸

Sollte es dennoch notwendig sein, dem Anwender die Möglichkeit einzuräumen, andere Datensätze, als die gerade angezeigten, verändern zu können, wird dazu extra im Formular selbst ein Button bereitgestellt, damit der Anwender nicht mit dem verdeckten Hauptmenü arbeiten muß. Dies ist z.B. bei der Erstellung eines Auftrages der Fall. Der Benutzer erhält hier u.a. die Möglichkeit, neue individuelle Textbausteine zu erstellen. Die Einführung einer zusätzlichen Schaltfläche ermöglicht es, nach Abschluß der Eingabe das aufrufende Formular per „Requery“ zu aktualisieren, damit der neue Baustein auch zur Auswahl bereit steht.

Dieser „Requery“ muß jedoch auf jeden Fall von dem aufgerufenen Form aus eingeleitet werden. Dies liegt an dem auf Multithreading ausgelegten Aufbau von Access, wodurch die aufrufende Routine eines Formulars nicht auf den Abschluß des aufgerufenen Formulars wartet, sondern, wie unter Multitasking üblich, weiter läuft und somit der Befehl „Re

²⁶ der dritte und zweite Knopf von rechts in der Fensterleiste mit dem Unterstrich- bzw. dem Kästchen-Symbol

²⁷ der ganz rechte Knopf in der Fensterleiste mit dem X-Symbol

²⁸ es hat sich gezeigt, daß der Durchschnittsanwender die angezeigten Forms nicht auf dem Desktop verschiebt, um verdeckte Forms wieder sichtbar und damit zugänglich zu machen.

query“ zu früh kommen würde. Da es jedoch unter Access nur möglich ist, über den Namen des Formulars auf dieses zuzugreifen und auch andere Formulare das nachträgliche Erfassen von individuellen Textbausteinen ermöglichen, wodurch der Name des zu aktualisierenden Formulars nicht immer gleich ist, muß man sich hier eines Tricks bedienen.

Jedes Formular erhält bei Aufruf eine Nummer, die fortlaufend ist. Damit hat das aufrufende Form immer den Index des aktuellen Forms minus eins. Schließt der Anwender nach dem Erfassen neuer Textbausteine das dazu eröffnete Formular über den hierzu eingeführten Schließen-Button, wird vor dem eigentlichen Schließen das aufrufende Form über die Funktion „Requery_Previous_Form“ aktualisiert.

Abbildung 8: Codeauszug „Nummer des aufrufenden Forms ermitteln“

```
Option Compare Database
Option Explicit

Public Function Index_Of_Form() As Integer
    Dim frm As Form, ctl As Control
    Dim index As Integer

    'Forms-Auflistung durchlaufen.
    index = 0
    For Each frm In Forms
        index = index + 1
    Next frm

    Index_Of_Form = index - 1 'Index ist immer einen zu groß
End Function

Public Sub Requery_Previous_Form()
    Dim index As Integer

    index = Index_Of_Form 'Aktuellen Index ermitteln.

    If (index > 0) Then 'Nicht, wenn einziges Form, neben Main
        Forms(index - 1).Requery 'Vorheriges Form aktualisieren
    End If
End Sub
```

Quelle: eigene Implementierung in Visual Basic, Codeauszug aus dem AES

Damit ist es auch klar, daß der Aufruf der Funktion nicht vom Hauptmenü aus erfolgen kann, da dann nur dieses aktualisiert werden würde, und nicht das Formular, welches auf die Aktualisierung angewiesen ist. Da ein Requery u.U. sehr zeitaufwendig ist, sollten auch nicht einfach alle offenen Formulare durchweg aktualisiert werden, sondern nur genau das eine, welches die Aktualisierung benötigt.

4.2.2 Aufbau eines Dialoges am Beispiel der Auftragsänderung

Der Benutzerdialog zur Auftragserfassung oder zur Auftragsänderung stellt den komplexesten Dialog dar, der für das AES entwickeln werden muß. Er ist zudem der wichtigste Dialog, da er die Hauptaufgabe des AES übernimmt. Daher muß bei der Erstellung ein wesentlicher Teil der Entwicklung darauf verwendet werden.

Abbildung 9: Beispielformular „Auftragsänderung“

Quelle: Screenshot aus Access 97, Formular zur Auftragsänderung

Wie man in Abbildung 9 erkennen kann, wurde ein neues und einheitlich Layout für die Fenster im AES gewählt. Der eigentliche Schließen-Button wurde deaktiviert und durch ein neues Icon ersetzt. Auch das Löschen eines Datensatzes wird einheitlich durch das Icon mit dem Mülleimer dargestellt. Jedes Formular erhält neben dem Text in der blauen Kopfleiste zusätzlich eine Überschrift neben den neu eingeführten Icons.

Um einen Auftrag zu erstellen oder zu ändern, muß der Benutzer alle Textbausteine zur aktuellen Auftragsart einsehen und entsprechende auswählen können. Ist ein Textbaustein noch nicht vorhanden, muß es möglich sein, einen neuen individuellen Textbaustein anlegen zu können. Auch müssen an dieser Stelle nachträglich neue Auftragsarten mit neuen

Textbausteinen erfaßbar sein. Ferner muß der Anwender auf die Reihenfolge der Textbausteine im Dokument Einfluß nehmen und falsche Textbausteine aus dem Auftrag wieder entfernen können. Dazu ist es hilfreich, wenn er den resultierenden Auftrag im Vorfeld als Druckvorschau betrachten kann.

Bei dem Design des Formulars war es ursprünglich geplant, mit Hilfe von Drag and Drop²⁹ und dem Prinzip des WYSIWYG³⁰ dem Benutzer eine komfortable Eingabe und Auswahl der Textbausteine, inklusive der Anordnung derselben im Text, zu bieten. Leider ist dies unter Access nicht oder nur mit großem Aufwand möglich.

Somit muß ein Formular entstehen, welches ähnlich komfortabel ist, jedoch nicht zu gewaltig und kompliziert wird. Zu diesem Zweck werden zwei Unterformulare eingeführt. Das eine zeigt die aktuellen Textbausteine der gewählten Auftragsart an, das andere die daraus bereits ausgewählten Bausteine inklusive einer Positionsnummer, nach der hier sortiert wird.

Der Anwender kann zyklisch durch die Textbausteine mit der Hilfe des Datensatznavigators wandern. Findet er einen zu übernehmenden Textbaustein, muß er auf das Icon mit dem Stift im darunter liegenden Unterformular klicken, damit dieser dann in den Auftrag übernommen wird. Durch Klicken auf das Icon mit dem roten Kreuz kann er einen erfaßten bzw. gerade aktiven Baustein wieder aus dem Auftrag entfernen.

Sollte der Text eines Bausteins größer sein, als der im Dialog zur Verfügung stehende Platz, kann der Anwender durch Klicken in den Text des Bausteins einen Rollbalken am rechten Rand erscheinen lassen, mit dessen Hilfe dann auch der Rest des Textbausteines sichtbar wird. Da diese Rollbalken unter Access nur in Textfeldern bei der Eingabe erscheinen, muß das Feld für Änderungen gesperrt werden, damit der Anwender nicht versehentlich den Text eines Bausteins bei dieser Aktion verändert.

Durch das Anpassen der Positionsnummern kann der Benutzer Einfluß auf die Reihenfolge der Bausteine im Dokument nehmen. Sollten Positionsnummern dadurch doppelt vorkommen, stellt dieses kein Problem dar. Eine Plausibilitätsprüfung der Nummern fehlt daher.

²⁹ s. Glossar

³⁰ Abkürzung für „What You See Is What You Get“, s.a. Glossar

Klickt der Benutzer auf das Icon mit der Lupe über dem Zettel, wird der Auftrag in seiner derzeitigen Form zur Ansicht gebracht und kann zum Test auch gedruckt werden. Mit dieser Hilfe ist man bei der Erstellung eines Auftrages nahe an der Technik des WYSIWYG.

Ferner hat der Anwender hier die Möglichkeit, die Druckmarkierung eines Auftrages wieder zu entfernen, damit dieser erneut gedruckt werden kann. Dazu muß er das Häkchen neben „Auftrag gedruckt.“ entfernen. Wird der Auftrag verändert, wird es automatisch gelöscht.

Der Button „Aktualisieren“ mußte wegen des Multithreadings von Access eingeführt werden. Normalerweise aktualisiert sich das Form nach der Erfassung neuer individueller Textbausteine nach der in Kapitel 4.2.1 bereits beschriebenen Technik selbständig. Doch es kann vorkommen, daß diese Aktualisierung nicht den gewünschten Erfolg bringt. Mit diesem Button hat der Anwender immer die Möglichkeit, die Datensätze des gesamten Formulars neu einzulesen und somit die Gewähr, jederzeit mit aktuellen Datensätzen zu arbeiten.

4.3 Drucken im AES

Um dem Anwender das Ausdrucken sämtlicher Formulare zu vereinfachen, wurde ein Dialog erstellt, der alle zu einem Mandanten existierende und noch nicht gedruckte Aufträge, sowie das ZMSD-Anmeldeformular und die Einzugsermächtigung, falls vom Mandanten gewünscht, anzeigt und zur Auswahl anbietet. Bereits gedruckte Aufträge erscheinen in diesem Dialog nicht mehr, können aber erneut aktiviert werden, wenn der Auftrag in dem Formular zur Auftragsänderung wieder zum Drucken freigegeben wird.

Neben den eigentlichen Aufträgen sind, der Anforderungsdefinition entsprechend, ein Formular zum Bankeinzug und das ZMSD-Formular zu drucken. Aber auch Berichte für allgemeine Informationen sind zu erstellen. So kann z.B. eine Übersicht über alle erfaßten Mandanten sowie alle Textbausteine, nach Auftragsarten sortiert, ausgedruckt werden.

Bevor ein Auftrag gedruckt werden kann, muß Access erst alle dazu nötigen Datensätze zusammenstellen. Dies wird ähnlich wie bei der Datenermittlung bei Formularen realisiert. Dazu erstellt der Programmierer im ersten Schritt eine Abfrage, die alle zur Auftragsnum

mer gehörende Textbausteine ermittelt und nach der Positionsnummer sortiert in einem Pool bereitstellt.

Danach wird der Bericht designed. Ein Bericht besteht aus einem Berichtskopf und Berichtsfuß, einer Kopf- und Fußzeile und einem Datenteil. Im Datenteil werden Platzhalter positioniert, die hinterher die entsprechenden Daten aufnehmen. Der Datenteil wird bei dem späteren Ausruck so oft wiederholt, wie es Datensätze gibt. Findet dabei ein Seitenumbruch statt, werden Kopf- und Fußzeile erneut gedruckt. Der Berichtskopf wird dagegen nur einmal auf der ersten und der Berichtsfuß nur ein einziges mal auf der letzten Seite gedruckt.

Abbildung 10: Das Druckmenü

The screenshot shows a dialog box titled "Mandant für Formulardruck wählen". It has a header bar with "Mandant auswählen" and icons for printer, stop, and print. Below the header are several input fields:

- Name wählen: Mustermann
- Mandantennummer: keine
- Name des Mandanten: Mustermann
- Vorname des Mandanten: Heiner
- Name der Firma: Testfirma ABC GmbH
- Straßenname: Musterweg 1
- Postleitzahl: 12345
- Ort: Musterdorf

Below these fields are two buttons: "Einzugsermächtigung drucken" and "ZMSD drucken".

There is a section titled "Mögliche Aufträge" with a sub-header "Aufträge zum Drucken auswählen". It contains a list of orders:

Order ID	Order Name
<input checked="" type="checkbox"/> 33/0000000008	Finanzbuchhaltung
<input type="checkbox"/> 29/0000000009	Teilnahme an Prüfungen

At the bottom, there is a "Datensatz:" field showing "9" and "von 9".

Quelle: Screenshot aus dem AES, Druckmenü mit eingblendetem Tipp-Text

Damit zusammenhängende Daten bei einem Seitenumbruch nicht getrennt werden, ist es möglich, diese als zusammengehörig zu definieren. Access druckt dann den Datensatz, sollte er nicht mehr auf eine Seite passen, komplett auf eine neue Seite.

Ist der Anwender mit seiner Auswahl im Druckmenü zufrieden, werden durch Klicken auf den Button mit dem Druckersymbol alle markierten Aufträge gedruckt. Dazu wird ein Formular geöffnet, welches auf einer Abfrage basiert, die Mandantennummer und Auf

tragsnummer aller zum Drucken markierter Aufträge zusammenstellt. Nach dem Drucken aller Aufträge wird das Formular automatisch wieder geschlossen, da es nur als Dummy verwendet wird, um die entsprechenden Aufträge mittels einer Abfrage auszuwählen und in einer temporären Tabelle zur Verfügung zu stellen.

Abbildung 11: Codeauszug „Alle Aufträge drucken“

```

Option Compare Database
Option Explicit

Private Sub Form_Load()

    If Me.CurrentRecord > 0 Then
        DoCmd.GoToRecord acDataForm, "Print_ausgewählte", acLast
        Do
            [Forms]![Menü_Start]![Returnvalue] = Me!Mandant
            [Forms]![Menü_Start]![Value_Auftragsnummer] = Me!Auftragsnummer

            DoCmd.OpenReport "Ein Auftrag", acViewNormal
            Me![Auftrag gedruckt?] = True
            Me![Zum Drucken ausgewählt] = False
            'MsgBox Me!Auftragsnummer
            If Me.CurrentRecord > 1 Then
                DoCmd.GoToRecord acDataForm, "Print_ausgewählte", acPrevious
            Else
                Exit Do
            End If
        Loop
    End If

    DoCmd.Close acForm, "Print_ausgewählte"

End Sub

```

Quelle: eigene Implementierung in Visual Basic, Codeauszug aus dem AES

Bevor das Formular zur Anzeige gebracht wird, führt Access selbständig eine Routine aus, die „Form_Load“ heißt. In dieser Routine werden nun alle durch die Abfrage ermittelten Aufträge in einer Schleife mit dem Bericht „Ein Auftrag“ von hinten nach vorne gedruckt und die beiden booleschen Attribute „Auftrag gedruckt?“ und „Zum Drucken ausgewählt“ entsprechend zurückgesetzt.

Damit der Bericht den richtigen Auftrag druckt, müssen die Mandantenummer und die Auftragsnummer an den Bericht übergeben werden. Da unter Access keine Parameterübergabe beim Aufruf eines Formulars oder Berichtes möglich ist, werden die Daten vor dem Aufruf des Berichtes im Hauptmenü in zwei nicht sichtbare Felder gespeichert, da das Hauptmenü immer geöffnet ist und von jedem Formular oder Bericht aus angesprochen werden kann.

5 Testphase

Ein Großteil der bei der Entwicklung von Software verwendeten Zeit entfällt auf das Testen³¹. Dabei stehen dem Programmierer mehrere Möglichkeiten und Techniken zum Testen zur Verfügung. Die am häufigsten angewandte Methode ist das White-Box-Testing, welches bereits während des iterativen Implementierungsprozesses regelmäßig Anwendung findet. Hierbei überprüft der Entwickler frisch implementierte Funktionen oder Routinen auf ihre Tauglichkeit und die Korrektheit des verwendeten Algorithmus.

Dazu wird eine Testumgebung um die implementierte Funktion erzeugt, welche die Eingabe- und Ausgabedaten der Funktion anzeigt oder speichert. Dabei sind die Methoden der Äquivalenzklassenbildung und der Grenzwertanalyse erste gute Ansätze. Da es sich um ein Programm auf der Basis eines DBMS handelt, sind die Testdaten für die Äquivalenzklassenbildung und der Grenzwertanalyse in Form von Datensätzen anzulegen.

Des Weiteren kann der Entwickler sein Programm mit möglichst realen Datenbeständen testen. Hierbei empfiehlt es sich, neben den für die Äquivalenzklassen und Grenzwerte verwendeten Datensätzen, weitere reale und auch defekte Datensätze mit widersprüchlichen Inhalten auszusuchen. Dadurch kann die Absturzicherheit und die Ausgabe von Fehlermeldungen überprüft werden.

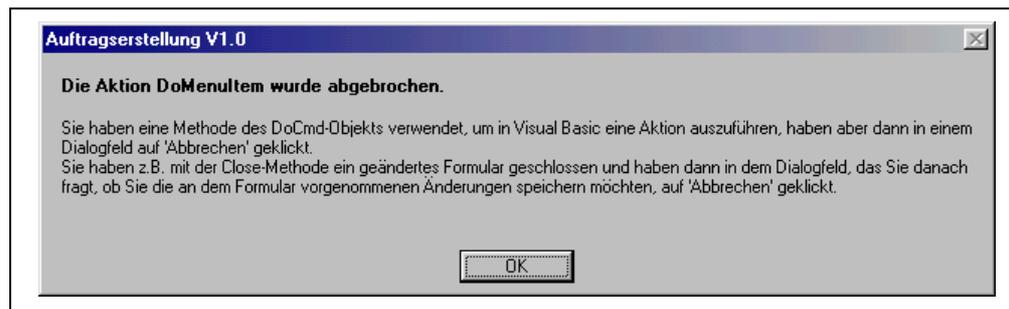
Neben dem White-Box-Testing, welches vornehmlich vom Entwickler durchgeführt wird, kann auch das Black-Box-Testing angewendet werden. Hierbei werden nicht die verwendeten Algorithmen auf bekannte Schwachstellen hin untersucht, sondern das gesamte Programm als Black-Box angesehen, welches Eingaben erhält und Ausgaben zurückgibt. Es wird dann überprüft, ob Ein- und Ausgabe zusammenpaßt und der Erwartung entspricht. Sehr leicht kann dieses Testverfahren vom späteren Anwender durchgeführt werden. Dazu sollte jedoch vorher ein Testkatalog aufgestellt werden, in welchem die durchzuführenden Testschritte festgehalten sind.

Bei dem Black-Box-Testing kann die Software zudem durch gleichzeitiges Anwenden von zwei oder mehr Benutzern sehr gut auf ihre Netzeigenschaften hin überprüft werden. Dazu sind im Testkatalog zeitlich abgestimmte Schritte aufzuführen, so daß z.B. gleichzei-

³¹ vgl. DUDEN Informatik, 1993, S.720f., Stichwort „Testen“

tig die Stammdaten eines Mandanten verändert und an anderer Stelle zugehörige Aufträge dieses Mandanten angelegt oder gelöscht werden. Ebenso muß überprüft werden, wie sich das Programm verhält, sollte ein Mitarbeiter einen Mandanten komplett löschen, während ein anderer gerade dessen Aufträge verändert.

Abbildung 12: Beispiel einer von Access automatisch erzeugten Fehlermeldung



Quelle: Screenshot aus Access 97, Fehlermeldung nach Löschabbruch

Neben den VB-Modulen muß der Programmierer auch die von ihm erstellten Formulare, Abfragen und Berichte testen. Das Testen von Abfragen ist recht einfach, da hier immer dann von einer korrekten SQL-Anweisung ausgegangen werden kann, wenn alle auf die Abfrage passenden Datensätze tatsächlich gefunden werden. Hierzu können gut die für das Testen der VB-Module eingeführten Datensätze weitere Verwendung finden.

Ähnliches gilt für die Berichte. Hierbei muß zusätzlich überprüft werden, ob das Dokument lesbar und aussagekräftig ist, und ob es mit dem verwendeten Briefkopf zu dem äußeren Erscheinungsbild der Kanzlei paßt.

Bei den Formularen ist vor allem die Fragestellung zu erörtern, ob die Dialogelemente sinnvoll angeordnet wurden und ob sie alle durch die Tabulatortaste in der richtigen Reihenfolge erreichbar sind. Zusätzlich müssen die Eingabefelder auf ihre Reaktion bei falschen Eingaben hin überprüft werden. So sollte ein rein numerisches Feld eingegebene Alphazeichen mit einer sinnvollen Fehlermeldung abweisen. Auch müssen sämtliche im Formular enthaltenen Hilfestellungen, Arbeitsanweisungen und Fehlermeldungen für den späteren Anwender verständlich sein. Ein schlechtes Beispiel stellt die in Abbildung 12 dargestellte und von Access automatisch ausgegebene Fehlermeldung dar.

6 Zusammenfassung und Ausblick

Das vorliegende Programm ist in seiner derzeitigen Version bereits einsetzbar. Leider sind noch nicht alle geplanten Funktionen implementiert worden. Vor allem bei der Datenschnittstelle zur ZMSD liegt dies u.a. daran, daß die Entwickler bei der DATEV keine Möglichkeit schufen, den Datenbestand ihrer Software durch Fremdsoftware verändern zu lassen.

Dadurch ist das bisher eingeführte ZMSD-Anlageformular, welches der Benutzer nach einer Neuanlage ausdrucken kann, der einzige Weg, die Übernahme des Mandanten in die ZMSD zu gewährleisten. Dies ist bisher die beste Lösung, da das AES nicht alle Daten erfaßt, die für die ZMSD benötigt werden. Somit müßte auch nach einer automatischen Übernahme der Daten eines Neumandates in die ZMSD ein Mitarbeiter den Datensatz manuell überprüfen und anpassen. Zur Erleichterung wurde auf dem ZMSD-Anlageformular Platz für einige der Daten gelassen, die vom Mandanten im Vorfeld zu erfragen sind und dort nach dem Ausdruck des Formulars eingetragen werden können.

Im Gegensatz ist es jedoch möglich, mit der Hilfe des ZMSD-Auskunftssystems Daten aus der ZMSD zu exportieren. Da hierbei äußerste Sorgfalt angewandt werden muß und vor allem die Mitarbeiter nicht überfordert werden dürfen, wartet auch diese Funktion noch auf ihre Implementierung.

Ferner ist die vorläufige Preiskalkulation eines Auftrages noch zu entwickeln. Dies ist bisher nicht geschehen, da das Programm auch ohne diese Funktion einsetzbar ist, die Implementierung jedoch sehr aufwendig zu sein scheint.

Auch ist im Druckbereich ein Fehler enthalten, der noch nicht behoben werden konnte. Wird ein Bericht gedruckt, dessen Textbausteine eine Seite komplett füllen, so wird die Unterschriftenzeile allein auf die nächste Seite gedruckt. Bisher ist es mir noch nicht gelungen, eine entsprechende Lösung unter Access zu finden. Denkbar wäre eine automatische Nachformatierung der Druckvorlage durch ein VB-Programm, welches auf genau diesen Fehler reagiert und dann Leerzeilen an einer entsprechenden Stelle einfügt, um den letzten Textbaustein auf die nächste Seite zu zwingen.

Des Weiteren ist es offensichtlich nicht möglich, von Access erzeugte Meldungen abzufangen und durch eigene Texte zu ersetzen. Meldungen, wie in Abbildung 12 zu sehen,

verwirren den Anwender, der nicht wissen kann, daß er ihnen keine weitere Beachtung schenken muß. Vor allem durch die eingeführte referentielle Integrität werden beim Löschen von Datensätzen automatisch Warnmeldungen von Access erzeugt. Der Inhalt dieser Meldung ist für einen Laien jedoch nicht verständlich.

Ansonsten wird das Programm in seiner Anwendung letztendlich zeigen können, was es zu leisten vermag und wo noch Schwachstellen auszuräumen sind.

Quellenverzeichnis

a) Monographien

- Barz, D. / Wunram, S. (1999), Mehr Honorar – aber wie?, Seminarunterlage des DATEV-KOLLEG „CHEF-Seminar“, Nr. 06/99-13 089-1, Nürnberg 1999.
- Becker, J. (1997), Skript zur Vorlesung Datenmanagement, Sommersemester 1997, Fachbereich Wirtschaftsinformatik, Westfälische Wilhelms Universität (WWU), Münster
- Hansen, H. R. (1998), Wirtschaftsinformatik I, Grundlagen betrieblicher Informationsverarbeitung, Bea, F.X., Dichtel, E., Schweitzer, M. (Hrsg.), 7., völl. neubearb. u. stark erw. Aufl. (1996), durchgeseh. Nachdruck (1998), Lucius & Lucius Stuttgart, 1998

b) Nachschlagewerke

- Claus, V. / Schwill, A. (1993), DUDEN Informatik, Ein Sachlexikon für Studium und Beruf, Engesser, H. (Hrsg.), 2., vollst. überarb. u. erw. Aufl., DUDENVERLAG Mannheim; Leipzig; Wien; Zürich, 1993

c) Elektronische Textmedien

- Access Hilfe (1997), Datei „ACMAIN80.HLP“, Version vom 04.09.1997

d) Gesetzestexte

- StBGebV (1998), Steuerberatergebührenverordnung v. 27.08.1998. Herausgegeben von: DATEV eG, 8., neu bearbeitete Auflage, Nürnberg, 1998

e) Gesprächsnotizen

- Becker, W. (2000), Gespräch mit Herrn Winfried Becker am 15.01.2000

Anhangsverzeichnis

Anhang	Seite
Anhang 1: Gesprächsnotiz zum Gespräch mit Herrn Winfried Becker	35
Anhang 2: Auszug aus der StBGebV, am Beispiel von §35	35
Anhang 3: Beispiel eines Auftrages	36

Anhang 3: Beispiel eines Auftrages

<p>Mustermann, Heiner Testfirma ABC GmbH Musterweg 1 12345 Musterdorf</p>	 <p>Winfried Becker Vereidigter Buchprüfer Steuerberater Rechtsbeistand</p> <p>Wirtschaftsberatung Investitionen Unternehmensberatung Existenzgründung 15.03.2000 <i>keine</i></p>	
<p>Auftragsnummer: 33/ 000000007 Auftragsart: Finanzbuchhaltung</p>		
<p>Position 1 Buchführung einschließlich des Kontierens der Belege</p>		
<p>Position 2 Laufende Überwachung der Buchführung des Auftraggebers</p>		
<p>Position 3 Hilfeleistung bei sonstigen Tätigkeiten im Zusammenhang mit der Buchführung</p>		
<p>_____ Unterschrift des Auftraggebers</p>		
<p>Vertragsbuchstelle KH Lippe Haus des Handwerkes Lemgo</p>	<p>PLZ 32631 = Postfach 124 PLZ 32657 = Engelbert-Kämpfer-Str. 4 Telefon (0 52 61) 93 714 Telefax (0 52 61) 93 71-71</p>	<p>Völkbank Lemgo 9 480 800 (BLZ 476 800 80) Sparkasse Lemgo 78 588 (BLZ 482 501 10)</p>
<p>Seite 1 von 1</p>		

Ehrenwörtliche Erklärung

Ich versichere, daß ich die beiliegende Projektarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Lemgo, 11. Januar 2002

Ort, Datum

Unterschrift