2.0 PROGRAMMENTWICKLUNG

2.1 Hardware

Zur Programmentwicklung stand uns die IBM-Großrechenanlage 3081 des Bayerischen Staatsministeriums für Ernährung Landwirtschaft und Forsten zur Verfügung. Diese ist auch als externer Rechner an das Bildschirmtextsystem angeschlossen. Über eine Datenleitung wird die Verbindung zu einer Steuereinheit IBM-3274 hergestellt, die mit mehreren Bildschirmen vom Typ 3278 verbunden ist. Ein LOEWE Editierplatz ermöglicht zusammen mit einem IBM-AT das Editieren von Btx-Seiten. Zur Programmbenutzung verwenden wir das Btx-Abfragegerät T3110 von Siemens.

2.2 Softwareeinsatz

Die IBM-Großenrechenanlage vom Typ 3081 arbeitet unter dem Betriebssystem OS/VS mit TSO und dem Editor ISPF. Die Btx-Dialogprogramme sind in der höheren Programmiersprache PL1 geschrieben und mit dem PL1-Optimizing-Compiler compiliert worden. Der Softwaretool MFS (Message Format Services) unterstützte die Input/Output Programmierung.
Als Datenbank und Datenkommunikationssystem kam IMS (Information Management System) von IBM zum Einsatz. Dieses bietet neben einer hierarchischen Datenbankverwaltung innerhalb des Datenkommunikationsteils auch eine Schnittstelle zum Btx-System und entsprach damit unseren Anforderungen. Diese Schnittstelle wird über die Bildschirmtextsoftware VCMIP von IBM realisiert. Als Subsystem von IMS enthält sie Routinen für die Erstellung und Speicherung von Btx-Seiten sowie Schnittstellen zu den Programmiersprachen Cobol, PL1 und Assembler.
Zum interaktiven Test von IMS-Dialogprogrammen stand BTS II (Batch Terminal Simulation) zur Verfügung. Um auftretende Probleme der Datenselektion zu lösen, arbeiteten wir mit den Programmen STAIRS/VS und STAIRS/MIKE, die im Rechenzentrum installiert sind.

2.2.1 Programmiersprache

Als Programmiersprache wählten wir die höhere Sprache PL1. Zum einen verfügten wir im Institut über umfangreiche Erfahrungen in der PL1 Programmierung, zum anderen waren folgende Eigenschaften dieser Sprache für die Realisierung des Projektes wichtig:

- Schnittstelle zu IMS
- Schnittstelle zur Btx-Software
- Schnittstelle zu STAIRS/MIKE
- Umfangreiche Hilfsfunktionen (z.B. translate)
- Gute Strukturierung von Programmen

- Übersichtliche Programmgestaltung
- Weitreichende Routinen für Fehlerbehandlung
- Komfortable Handhabung des Compilers, besonders gute Beschreibung von Codierungsfehlern

Mit dem Einsatz von PL1 ergaben sich während der Projektlaufzeit keine Probleme.

2.2.2 Datenbanksoftware

Das Datenmanagementprogramm IMS (Information Management System) besteht aus zwei Komponenten, dem Datenbankteil und dem Datenkommunikationsteil.
Der Datenbankteil umfaßt eine Reihe von Datenbanktypen, die alle zur Kategorie der hierarchischen Datenbanken gehören. Ihr gemeinsames Merkmal sind die hierarchischen Beziehungen der einzelnen Segmente untereinander. Zum besseren Verständnis sind im folgenden die grundsätzlichen Strukturelemente einer IMS-Datenbank zusammengefaßt.

1. Datenbanksatz (data base record)

   Es handelt sich dabei um alle Informationen, die zu einem bestimmten Schlüsselbegriff gehören. Wenn beispielsweise die Betriebsnummer in einer Schlagkarteidatenbank der Schlüsselbegriff ist, dann besteht der Datenbanksatz aus der Summe der Informationen über sämtliche Schläge, wie Bodenbeschreibung, Anbau-, Düngungs-, Pflanzenschutz- und Erntemaßnahmen.

2. Segmenttyp (segment type)

   Eine Teilmenge des Datenbanksatzes, die Informationen zu einem bestimmten Thema in einem festgelegten Format enthält, heißt Segment. Man könnte zum Beispiel ein Segment für die Düngung festlegen, in dem jeweils die Art des Düngers, die Inhaltsstoffe und die ausgebrachte Menge festgehalten werden.

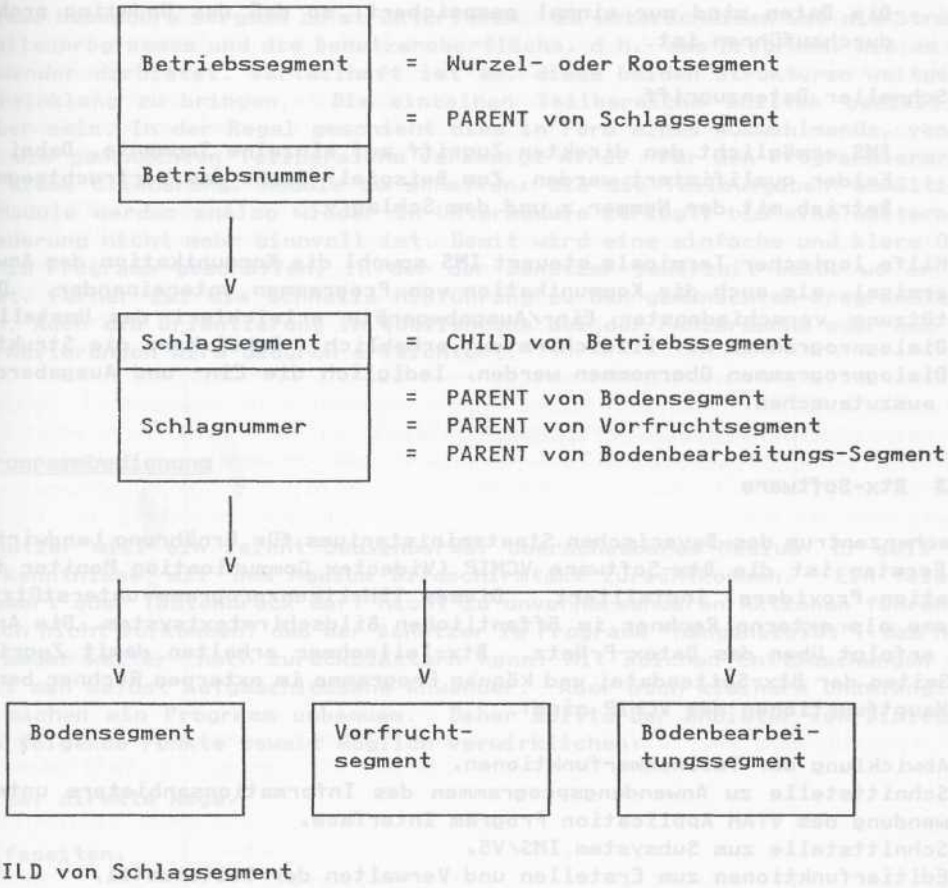
3. Segmentvorkommen (segment occurence)

   Sollte mehr als eine Düngungsmaßnahme durchzuführen sein, stellt jeweils eine Maßnahme ein Segmentvorkommen dar.

4. Feld (field)

   Ein Feld bezeichnet die kleinste ansprechbare Informationseinheit in einem Segment. Das kann z.B. der Code für den Düngernamen sein. Es ist dabei möglich, nach Inhalten von Feldern zu selektieren.

Zur Illustration der hierarchischen Struktur wurde Übersicht 1 auf Seite 5 eingefügt; ein Auszug aus unserer Schlagkartei-Datenbank.



Übersicht 1. Hierarchische Beziehungen von Segmenten

In der obersten Hierarchiestufe steht das Betriebssegment. Dieses wird je Betrieb nur einmal angelegt. Für jeden Schlag gibt es dann ein Segmentvorkommen des Schlagsegments, das mit dem Betriebssegment in einer sogenannten PARENT-CHILD Beziehung steht. Das Schlagsegment ist dabei vom Betriebssegement abhängig. Weitere derartige Beziehungen ordnen die Inhalte der Segmente Boden, Vorfrucht und Bodenbearbeitung einem Schlag zu. Dadurch wird in der Datenbank eindeutig definiert, welche Informationen zu welchem Schlag gehören.
Für unserer Projekt ergaben sich dadurch folgende Vorteile:

1. Sparen von Speicherplatz

   Die Zahl der Segmente ist den jeweiligen Anforderungen angepaßt. Das heißt, bei Betrieben mit wenig Schlägen werden weniger Segmente angelegt, als bei Betrieben mit vielen.

2. Zugriff mehrerer Anwendungen auf dieselbe Datenbank

Die Daten sind nur einmal gespeichert, so daß das Updating problemlos durchzuführen ist.

3. Schneller Datenzugriff

IMS ermöglicht den direkten Zugriff auf einzelne Segmente. Dabei können Felder qualifiziert werden. Zum Beispiel: Suche das Vorfruchtsegment im Betrieb mit der Nummer x und dem Schlag y.

Mit Hilfe logischer Terminals steuert IMS sowohl die Kommunikation des Anwenders am Terminal, als auch die Kommunikation von Programmen untereinander. Die Unterstützung verschiedenster Ein-/Ausgabegeräte erleichtert das Umstellen von IMS-Dialogprogrammen auf Bildschirmtext erheblich. Dabei kann die Struktur von IMS-Dialogprogrammen übernommen werden, lediglich die Ein- und Ausgaberoutinen sind auszutauschen.

2.2.3 Btx-Software

Am Rechenzentrum des Bayerischen Staatsministeriums für Ernährung Landwirtschaft und Forsten ist die Btx-Software VCMIP (Videotex Communication Monitor for Information Providers) installiert. Dieses IBM-Lizenzprogramm unterstützt IBM-Systeme als externe Rechner im öffentlichen Bildschirmtextsystem. Die Ankoppelung erfolgt über das Datex-P-Netz. Btx-Teilnehmer erhalten damit Zugriff auf die Seiten der Btx-Seitendatei und können Programme im externen Rechner benutzen. Die Hauptfunktionen des VCMIP sind:

1. Abwicklung der Teilnehmerfunktionen,
2. Schnittstelle zu Anwendungsprogrammen des Informationsanbieters unter Verwendung des VTAM Application Program Interface,
3. Schnittstelle zum Subsystem IMS/VS,
4. Editierfunktionen zum Erstellen und Verwalten der Seitendatei,
5. Unterstützung von mehreren Informationsanbietern auf einem System,
6. Steuerfunktionen für die Bedienung,
7. Unterstützung des 24-Stunden-Betriebs.

Nach den üblichen Anfangsschwierigkeiten wurde die Software schnell stabil. Immer noch gibt es Probleme mit falsch editierten Seiten. Das sind z.B. Seiten deren Ein- oder Ausgabebereich die zulässige Größe überschreitet. Als Folge endet das Programm fehlerhaft. Ein Hinweis auf die Ursache des Fehlers wird nicht ausgegeben.

2.3 Programmstruktur

Bildschirmtext-Dialogprogramme haben denselben Ablauf wie 'normale' IMS-Dialogprogramme. Das herausragende Merkmal ist, daß das Programm bei jeder Ein-/Ausgabeaktion neu in eine IMS-Region geladen wird und die Inhalte von Variablen dabei verloren gehen. Nur der Inhalt eines speziellen Bereiches (SPA) wird zwischengespeichert. Im Unterschied zu 'normalen' IMS-Dialogprogrammen erfüllt bei unseren Bildschirmtext-Programmen eine Datenbank die Aufgabe als Zwischenspeicher. Da vor allem auch Laien mit den Btx-Dialogprogrammen zurechtkommen sollen,

sind diese besonders sorgsam zu strukturieren. Zu unterscheiden ist die Struktur des Quellenprogrammes und die Benutzeroberfläche, d.h. das Programm, wie es sich dem Anwender darbietet. Vorteilhaft ist es, diese beiden Strukturen weitgehend in Übereinklang zu bringen. Die einzelnen Teilbereiche sollten gezielt ansprechbar sein. In der Regel geschieht dies in Form eines Auswahlmenüs, von dem aus in die gewünschten Teilbereiche verzweigt wird. Für den Programmierer bedeutet diese Gliederung, Module zu schaffen, die die Teilaufgaben bewältigen. Diese Module werden analog wieder in Untermodule zerlegt, bis eine weitere Untergliederung nicht mehr sinnvoll ist. Damit wird eine einfache und klare Gliederung im Programm geschaffen, in der der Benutzer jederzeit weiß, wo er sich befindet. Ferner ist die schnelle Hinführung zu den gewünschten Programmteilen gegeben. Auch die Orientierung im Quellencode bei der Fehlersuche oder bei Programmerweiterungen wird dadurch erleichtert.

2.4 Programmbedienung

Der Benutzer will ein leicht bedienbares, überschaubares Medium. Er soll ohne EDV-Vorkenntnisse mit dem Medium Bildschirmtext zurechtkommen. Ein falscher Eingabewert oder Tastendruck darf nicht zu unvorhersehbaren Aktionen führen. Es darf auch nicht vorkommen, daß der Benutzer im Programm "hängenbleibt", das heißt daß er weder weiter- noch zurückblättern kann. Mit solchen Enttäuschungen verschreckt man selbst aufgeschlossene Anwender. Aber auch kleinere Unzulänglichkeiten machen ein Programm unbequem. Daher sollte der Anbieter von Dialogprogrammen folgende Punkte soweit möglich verwirklichen:

- Kurze, direkte Wege,
- Hilfeseiten,
- Kurze, klare Formulierungen,
- Sorgfältige Benutzerführung,
- Übersichtliche Programmgestaltung.

In vielen Fällen erweist sich eine gedruckte Programmbeschreibung als äußerst hilfreich. Besonders wo aus Platzgründen mit Abkürzungen gearbeitet wird - in unserem Schlagkarteiprogramm kürzen wir viele Angaben, z.B. Fruchtarten, Düngemittel und Pflanzenschutzmittel mit Schlüsseln ab - erleichtert eine Anleitung das Arbeiten.

2.5 Programmgestaltung

An den zwei Punkten Seitengestaltung und Programmsteuerzeichen soll gezeigt werden, wie wir versucht haben, den Forderungen der Programmbedienung durch die Programmgestaltung gerecht zu werden.

1. Seitengestaltung

Für Dialogprogramme kommt es mehr auf funktionelle Gestaltung als auf schöne Bilder an. Trotzdem sollten für das Auge angenehme Farben verwendet werden. Wir entschieden uns für grau als Hintergrundfarbe und weiße, gelbe und schwarze Schrift im Vordergrund. Die Überschrift in doppelter Größe und schwarzer Farbe bei grünem Hintergrund belegt die beiden obersten Zeilen einer Seite. Bis etwa zur zwanzigsten Zeile erstreckt sich der Hauptteil einer Seite. Je nach Zweck werden hier Informationen geboten, Daten erfaßt oder Ergebnisse ausgegeben. Die Texte sollten überschaubar gegliedert sein. In der meist weiß gehaltenen Schrift sind wichtige Passagen durch gelbe Farbe hervorgehoben. Eingabefelder werden durch Unterstreichungszeichen gekennzeichnet und mit klarer Beschriftung versehen. Gegebenenfalls sind sie obendrein durch Farbe hervorzuheben.

```
Überschrift (grüner Balken)
----------------------------------------

Erfassungsteil

        - Text, weiße Farbe

        - Eingabefelder, gelbe Farbe

----------------------------------------
Balken (grün)                Auswahl _
----------------------------------------
Hinweise zu Benutzerführung
```

Übersicht 2. Aufbau einer Schlagkartei-Erfassungsseite

Die zusätzliche Möglichkeit von Bildschirmtext, einzelne Eingabefelder mit Hinweisen (Promptingtexten) zu versehen, verwenden wir sehr sparsam, da es unbequem ist, mit den Augen bei jedem Eingabefeld aus dem Text in die letzte Zeile und dann wieder zurück zu springen. Zudem verlängern Promtingtexte die Übertragungszeit von Seiten und behindern das überspringen von Feldern, auf denen keine Angaben nötig sind. In den unteren Zeilen, vom Hauptteil durch einen grünen Balken getrennt (Übersicht 2), befinden sich die Hinweise zur Benutzerführung in schwarzer Schrift. Das Auswahlfeld steht immer ganz rechts im grünen Balken. Häufig wird kritisiert, daß Btx-Dialogprogramme besonders im Vergleich mit Programmen auf Personal Computern zuviel Zeit in Anspruch nehmen. Unter diesem Gesichtspunkt ist möglichst viel Inhalt in eine Seite zu packen, um mit möglichst wenig Seiten auszukommen. Andererseits sollen die Seiten auch nicht überladen sein, denn dadurch leidet die Übersichtlichkeit. Allgemeingültige Ratschläge für die Gestaltung des Seitenin-

haltes fallen wegen der Verschiedenartigkeit der Aufgaben in den einzelnen Programmen schwer. In jedem Fall kann man kurze und eindeutige Formulierungen empfehlen.

2. Programmsteuerzeichen
Der Benutzer soll durch das Programm "geführt" werden, was bedeutet, daß ihm an jeder Stelle mitgeteilt wird, welche Auswahlmöglichkeiten er besitzt. Die Benutzerführung soll in einheitlicher Weise erfolgen. Für Anwender, die zudem mit Personal-Computer oder Großrechner arbeiten, sollten Steuerzeichen so gewählt werden, daß sie den Programmfunktionstasten entsprechen. Diese Forderung konnte nicht vollständig erfüllt werden, da die Funktionen '0' für 'zurück' und '1' für 'weiter' allgemein anerkannter Btx-Standard sind. Obwohl verschiedene Programme unterschiedliche Steuerungselemente benötigen, haben wir doch versucht, in unseren Programmen gleiche Steuerzeichen für gleiche Funktionen zu verwenden: '0' entspricht zurück, '1' weiter, '4' verwenden wir für Speichern, '5' für Löschen, '7' für Rückwärtsblättern, '8' für Weiterblättern und '9' für Programmende. Allgemeine Informationen zum Programm können mit 'I', Hinweise zu den Eingaben auf einer bestimmten Seite mit 'H' eingeblendet werden. Die Funktionen '2', '3', und '6' können nach Bedarf verwendet werden.

2.6 Programmvernetzung

Innerhalb dieses Projektes sind, wie schon in der Einleitung erwähnt, die folgenden sechs Dialogprogramme für Btx entwickelt worden.

1. Schlagkartei (Erfassung und Auswertung),
2. Epipre,
3. Septoriaprognose,
4. Mischverträglichkeit von Pflanzenschutzmitteln,
5. Herbizidauswahl und
6. Düngeplanung

Schon vor der Realisierung einzelner Programme stellte sich die Frage, ob diese unabhängig laufen oder untereinander vernetzt werden sollten. Für den Entscheidungsprozeß sind zunächst die Vor- und Nachteile beider Möglichkeiten aufgezeigt worden.

Als Vorteil für die Unabhängigkeit der Programme untereinander und als Nachteil der Vernetzung ergibt sich:

- Einzelne Programme sind nicht auf die Funktionsfähigkeit anderer Programme angewiesen.
- Innerhalb der Programmierung kann für jedes Programm Struktur und Aufbau speziell auf das Programm ausgerichtet werden.

- Notwendige Modifikationen in einem Programm beeinflussen die übrigen Programme nicht.

Als Nachteil der Unabhängigkeit der Programme und als Vorteil für die Vernetzung ergibt sich:

- Gleiche Daten, die von mehreren Programmen benötigt werden, müssen vom Benutzer mehrmals für einzelne Programme eingegeben werden.
- Die Aktualisierug von Daten muß ebenfalls mehrmals erfolgen.
- Die Wartung des Gesamtpaketes wird schwierig, da u.U. in jedem einzelnen Programm eigene Konventionen gepflegt werden.
- Das Wechseln von einem Programm zum anderen ist zeitaufwendig, da zunächst die "Beendigungs-Prozedur" des einen Programmes durchgeführt und dann die "Anfangs-Prozedur" des nächsten Programmes durchlaufen werden muß.

Da es für beide Varianten sowohl Vorteile als auch Nachteile gibt, ist in unserem Fall eine Mischlösung realisiert worden. Diejenigen Programme, die auf gemeinsame Daten zugreifen und diese deshalb in einer gemeinsamen Datenbank abspeichern, sind vernetzt worden, während die übrigen unabhängig voneinander existieren. Zu den ersten gehören die Schlagkartei, die Düngeplanung und Epipre. Zu den letzteren zählen die Septoriaprognose, das Mischprogramm und die Herbizidauswahl. Während Schlagkartei und Epipre dieselben Datenerfassungsseiten benutzen, ist für die Düngeplanung parallel eine eigene Datenerfassung vorgesehen. Auch Benutzer, die keine Schlagkartei über Btx führen wollen, können auf diese Weise eine Düngeplanung rechnen. Die Datenspeicherung erfolgt in jedem Falle sowohl bei der Schlagkartei als auch bei der Düngeplanung in derselben Datenbank. Das bedeutet beispielsweise, daß Daten, die über das Schlagkarteiprogramm eingegeben wurden, in der Düngeplanung schon vorhanden sind und umgekehrt. Für den Benutzer sieht die Vernetzung so aus, daß er vom Hauptauswahlmenü Schlagkartei direkt Epipre und die Düngeplanung anwählen kann. Intern wird dabei das jeweils gewünschte Programm geladen und ausgeführt. Technisch ist dies mit Hilfe der eingesetzten Software ohne Probleme zu lösen.

Bei der Septoriaprognose, dem Mischprogramm und der Herbizidauswahl ist demgegenüber eine Vernetzung nicht sinnvoll, weil diese Programme mit unterschiedlichen Daten rechnen und dabei die einzugebende Datenmenge minimal ist. Daher kann der Benutzer mit ihnen schnell zu seinem Ergebnis kommen, ohne sich zuvor mit anderen Programmen beschäftigen zu müssen.

2.7 Datenschutz

Um mit Dialogprogrammen, wie beispielsweise der Schlagkartei oder dem Programm zur Düngeberechung, in Btx arbeiten zu können, gibt der Benutzer zuvor die für die Berechnung notwendigen Daten ein. Da der Rechner diese zunächst speichert, erspart sich der Benutzer bei weiteren Arbeiten eine erneute Eingabe. Lediglich die Korrektur früherer Daten ist durchzuführen. Dies verkürzt die Zeit der Programmanwendung und verringert damit auch die Btx-Telefonrechnung.

Diesem Vorteil stehen Probleme der Datensicherheit gegenüber: Da es sich um persönliche, betriebsspezifische Daten handelt, sind im Programm Maßnahmen zur Datensicherung eingebaut.

Ein wirksamer Schutz läßt sich über den kontrollierten Zugang zu Dialogprogrammen mit Hilfe einer Kombination aus Benutzernummer und Passwort erreichen. Während als Betriebsnummer eine amtliche (in Bayern die Gasölbetriebsnummer) einzugeben ist, ist das Passwort vom Benutzer frei wählbar und jederzeit zu ändern. Um mögliche 'Hacker'-Erfolge Dritter, d.h. Entdecken des Passwortes durch dauerndes Ausprobieren, zu verhindern, wird bei mehr als dreimaliger falscher Eingabe des Passwortes die Verbindung zwischen externem Rechner und Btx-System getrennt. Die völlige Sperrung einer Benutzernummer ist vorgesehen, wenn eine bestimmte Anzahl von falsch eingegebenen Passwörtern überschritten wurde.

Neben diesen Schutzmaßnahmen ermöglicht das Programm dem Benutzer seine eingegebenen Daten, die sonst bis zum nächsten Aufruf im externen Rechner bleiben, jederzeit selbst zu löschen. Der Vorteil dieser 'absoluten' Datensicherheit ist gegenüber dem Nachteil einer mehrmaligen Dateneingabe von jedem Benutzer selbst abzuwägen.

2.8 Programmtest

Das Kapitel Programmtest ist bei Btx-Dialogprogrammen ein besonderes Problem. Ein effizienter Programmtest über Btx ist kaum möglich, da das System dafür zu langsam ist und obendrein bei auftretenden Fehlern keine Fehlerhinweise gibt. Lediglich die Endkontrolle eines Programmes ist hier möglich, um ganz speziell Btx-Masken und evtl. Schnittstellenprobleme zwischen dem von uns verwandten Datenbanksystem und dem Btx-System zu überprüfen.

Aus diesem Grund ist ein gestaffeltes System von Prüfmechanismen und Hilfen entwickelt worden, um den grundlegenden Programmtest über den Großrechner abwickeln zu können.

2.8.1 Stabilitätsanforderungen

Grundsätzliches Ziel bei lauffähigen Programmen über Btx sollte es sein, daß diese Programme auf jede Situation der Benutzer richtig reagieren. Diese Optimalforderung kann sicherlich erst, wie bei anderen Programmen auch, nach einigen Jahren Praxistest hundertprozentig erreicht werden. Dialogprogramme sind so konzipiert, daß der Anwender sie an einer bestimmten Stelle starten und durch Eingabe eines zuvor definierten Steuerzeichens beenden kann. Verhält sich das Programm in jeder Situation so, spricht man von Programmstabilität.

Neben diesem normalen Kommunikationsablauf zwischen Programm und Benutzer können Dialogprogramme aber auch unkontrolliert, d.h. gegen den Willen des Anwenders enden, auch 'abstürzen' genannt. Dies führt zur Verunsicherung und zum Ärger, wenn beispielsweise durch den unkontrollierten Programmabbruch zuvor eingegebene Daten verloren gegangen sind.

Welche Maßnahmen können Instabilitäten des Programms verhindern:

- Prüfung von Eingabedaten,
- Abfangen von Fehlern in der Programmierung.

Prüfung von Eingabedaten

Ein Dialogprogramm benötigt Eingabedaten vom Benutzer, um daraus Ergebnisse berechnen zu können. Dazu muß der Programmanbieter Art und Umfang der einzugebenden Daten vorher festlegen, um den Auswertungsteil auf sie abstimmen zu können. Bei der Eingabe zugelassener Daten funktioniert dieses fehlerfrei. Nicht zugelassene Eingabedaten, z.B. Buchstaben statt Zahlen, würden die Programmstabilität gefährden. Daher ist es Aufgabe des Programmanbieters, die Eingabedaten nach den von ihm aufgestellten Kriterien zu prüfen und falsche Daten zurückzuweisen. Diese Forderung klingt zwar banal, in der Praxis bedarf es jedoch großen Aufwandes in der Programmierung, um zugelassene Daten von unzulässigen Daten zu trennen.

Für die Programmsteuerung ergibt sich somit folgender Ablauf:

- die Programmseite erscheint am Bildschirm,
- der Benutzer gibt seine Daten ein,
- er wählt anschließend die nächste Eingabeseite,
- das Programm prüft die eingegebenen Daten,
- falls die Prüfung erfolgreich war, erfolgt die Datenspeicherung,
- einblenden der nächsten Eingabeseite bzw. Ausgabe des Ergebnisses.

Durch diesen Ablauf ist gewährleistet, daß das Programm auf Grund fehlerhafter Eingabedaten nicht unkontrolliert endet.

Abfangen von Fehlern in der Programmierung

Neben unzulässiger Dateneingabe des Benutzers können auch Programmierfehler des Anbieters zum Abbruch führen. Gerade beim Ersteinsatz eines Programmes werden diese Fehler sichtbar, weil auf Grund des Praxiseinsatzes Situationen im Programm entstehen können, auf die dieses nicht wie geplant reagiert.

Das EDV-System setzt in solchen Fällen einen sogenannten Fehlercode, der je nach Schwere des Fehlers den Abbruch des Programmes auslöst. Viele Programmiersprachen bieten jedoch die Möglichkeit, diese Fehlercodes abzufragen, d.h. gezielt auf solche Situationen durch das Einleiten einer vom Programmierer vorbestimmten Aktion zu reagieren. Ein Programmabbruch findet dann nicht statt. Durch die Ausgabe der Fehlercodes läßt sich der Fehler genau lokalisieren, wenn der Benutzer sie dem Programmanbieter mitteilt.

2.8.2 Fehlerbehandlung

Grundsätzlich ist zwischen Meldungen von Programm- und Eingabefehlern zu unterscheiden.

Meldungen von Programmfehlern

Meldungen von Programmfehlern informieren den Benutzer über das Auftreten eines Programmfehlers, der durch fehlerhafte Programmierung aufgetreten ist. Diese Meldungen sollten im Interesse des Anbieters eine Telefonnummer zur Benachrichtigung bei Störungen enthalten. Darüberhinaus ist die Ausgabe eines Codes sinnvoll, aus dem der Programmanbieter bei einer telefonischen Störungsmeldung den Fehler schnell lokalisieren kann. Geeignet ist dazu der Name des Unterprogrammes, in dem der Fehler aufgetreten ist und der Wert wichtiger Zähl- und Steuervariablen. Nur durch präzise Fehlermeldungen kann der Programmanbieter Fehler grundlegend beseitigen. Als Beispiel sei die Programmfehlermeldung aus dem Düngeplanungsprogramm aufgeführt.

```
BALIS-TUM Weihenstephan
++++++++++++++++++++++++++++++++++++++
     Allgemeine Fehlerbehandlung

Im Dialogprogramm trat unerwartet
ein Fehler auf. Bitte notieren Sie
zur schnelleren Diagnose folgende
Werte:
Seitennummer  : ...............
Unterpgmnr    : ........
Fehlercode    : ....
Ausl. Zeichen : .
Ausl. String  : ...................
++++++++++++++++++++++++++ Auswahl => .
.......................................
9 << ENDE              weiter im Pgm => 1
```

Übersicht 3. Maske zur Ausgabe von Programmfehlern

Meldungen von Eingabefehlern

Sowohl unzulässige, als auch vom Wert her unwahrscheinliche Eingaben können Fehlermeldungen hervorrufen. Sie setzen sich je nach ihrem Aussagegehalt aus unterschiedlichen Komponenten zusammen. Diese sind

- Angabe 'Fehler',
- Zeilenangabe, in der Fehler auftreten,
- Spaltenangabe, in der Fehler auftreten,
- Angabe der Art des Fehlers,
- Hinweis zur weiteren Vorgehensweise.

Je mehr der obenstehenden Komponenten in Fehlermeldungen eines Programmes enthalten sind, um so leichter ist für den Benutzer die Lokalisierung eines Fehlers und dessen Behebung. Gleichzeitig nimmt damit aber auch der Programmieraufwand überproportional zu. Die meisten Programmanbieter sehen bisher die Fehlerkontrolle und ihre Meldung als notwendiges Übel an. Die Aussagekraft der Meldungen hat aber einen ganz entscheidenden Einfluß auf die Attraktivität des Programmes. Häufige Hinweise auf nichtssagende Fehlermeldungen lassen den am Anfang aufge-

schlossenen Benutzer schnell verzweifeln. Daher ist dieses Gebiet mit besonderer Sorgfalt zu behandeln.

Grundsätzlich ist immer darauf zu achten, Fehlermeldungen so zu gestalten, daß sie dem Programmbenutzer schnell und gezielt helfen können.

2.8.3 Parallel laufendes Testprogramm

Da Dialogprogramme im Btx-System auf Grund technischer Voraussetzungen wesentlich langsamer ablaufen als auf dem Personal-Computer oder Großrechner und zudem die Fehlersuche zu schwerfällig ist, kann ein effizienter Programmtest nur schwer erreicht werden. Unbedingt notwendig war deshalb zur Programmentwicklung ein parallel laufendes Testprogramm am Großrechner. Mit Hilfe des Softwarepaketes BTS (Batch Terminal Simulator) und dem Maskengenerator (MFS) ist es möglich, auf dem Großrechnerterminal eine 1:1-Abbildung der späteren Btx-Seiten zu editieren und mit BTS den Programmablauf zu simulieren. Das Entscheidende an diesem Testprogramm ist, daß das Hauptprogramm für Test und späteren Btx-Einsatz identisch ist. Lediglich verschiedene systembedingte Ausgabemodule für Test und Btx werden an das Hauptprogramm gekoppelt. Auf diese Weise konnten die Dialogprogramme zuerst in der Simulation entwickelt und durchgetestet werden. Danach folgte die Umstellung in das Btx-System. Diese dauert selbst für größere Programme wie Düngeplanung oder Schlagkartei nicht länger als eine Woche. In der anfänglichen Programmplanung ist jedoch jedes Programm so zu strukturieren, daß zwischen identischen Programmteilen und gesonderten Programmteilen für Btx und TEST unterschieden wird. Die Programme müssen dann so programmiert werden, daß jeweils unterschiedliche Programmteile als separate Programmodule ausgetauscht werden können. Erfahrungen während der Arbeit haben gezeigt, daß diese Programmplanung unbedingt notwendig ist, um spätere Umstellungen von Test auf Btx mit geringem Zeitaufwand durchführen zu können. Welche Vorteile bietet nun ein parallel laufendes Testprogramm dieser Art:

- Die Programmgeschwindigkeit ist um den Faktor 5 bis 10 höher.
- Das Testprogramm läuft unabhängig vom Btx-Programm.
- Bei auftretenden Fehlern im Btx-Programm kann parallel dazu getestet und der Fehler gesucht werden.
- Parallel zum Testprogramm wird von BTS ein Protokoll mitgeschrieben, anhand dessen nach der Sitzung alle Schritte nachvollzogen und überprüft werden können.
- Zwischen der Einblendung der einzelnen Eingabeseiten erfolgt eine Auflistung aller Datenbankzugriffe. Dadurch können schon viele Fehler entdeckt werden, die mit Datenzugriff und -abspeicherung zu tun haben.
- Die spätere Programmpflege wird sehr erleichtert, da vorgenommene Modifikationen zunächst ebenfalls in der Simulation laufen und dort getestet werden können, bevor sie in das Btx-System gestellt werden.