



SCHOTT

Hans Kneilmann
Prozeßrechner-Anwendung
Schott Glaswerke Mainz

Mainz, 6. Juli 1982
TDE-3/Knm/Mty - 2327

AUMADISY

Inhaltsverzeichnis

1. Einleitung
2. Systemübersicht
3. Gründe für eine Eigenentwicklung
4. Sprachumfang
5. Datenbereiche und Adressierungsvarianten
6. Beispiel Dialogsteuersatz
7. Zusammenfassung

1. Einleitung

AUMADISY ist ein anwendungsunabhängiges, maskenorientiertes Dialogsystem, zur Definition und Abarbeitung von rechnergeführten Dialogen am Bildschirm.

Es ermöglicht eine Dialogprogrammierung auf hohem Niveau, in der Art einer blockstrukturierten Programmiersprache.

Entwickelt wurde AUMADISY im Rahmen der Erstellung eines "Lager-Platz- und -Bestands-Verwaltungssystems".

Entwicklungsziele waren:

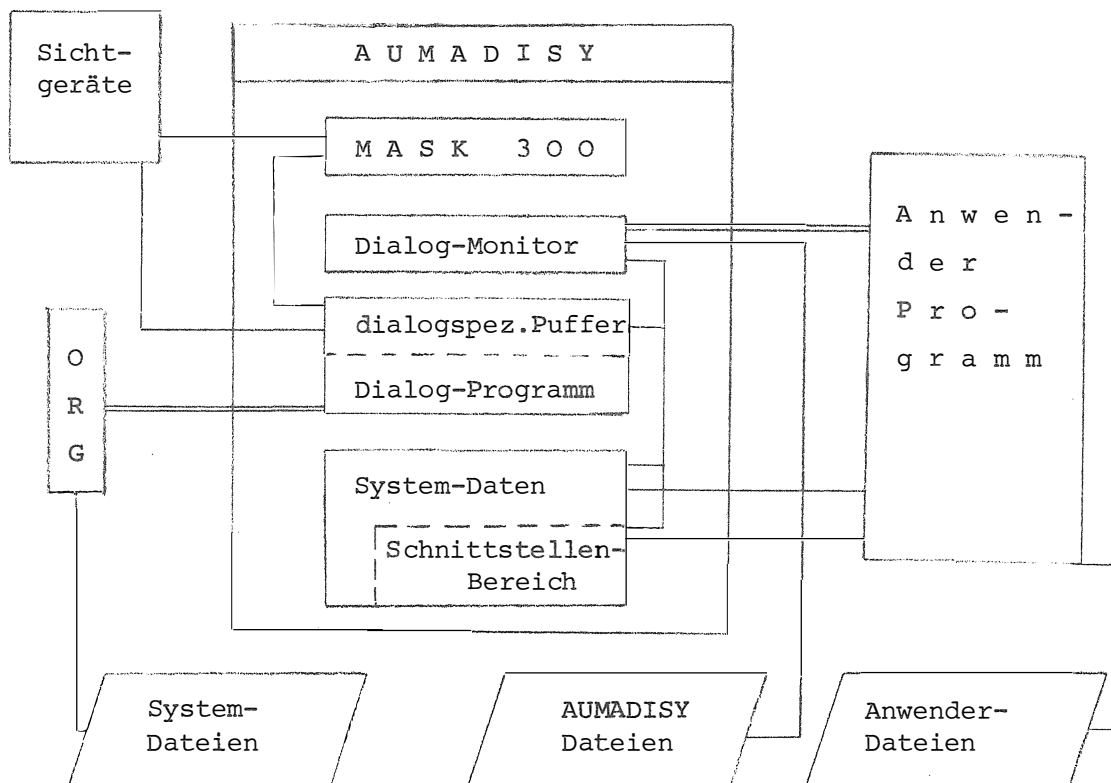
- Anwenderfreundlichkeit
- Mehrbenutzerbereich mit beliebig vielen Terminals
- Dialogteil universell einsetzbar.



Die Anwenderfreundlichkeit wurde u.a. erreicht durch:

- strenge Bedienerführung, trotz freilaufendem Cursor
- sofortige Kontrolle der Eingabe auf beliebige Kriterien
- Ausgabe von Hinweistexten, z.B. nach fehlerhaften Eingaben
- die tabellarische Ausgabe der Eingabemöglichkeiten
- die Möglichkeit, den Inhalt von Eingabefeldern für die Mehrfachbenutzung eines Bildes zu retten.

2. Systemübersicht



AUMADISY bildet die Schnittstelle zwischen dem Bediener am Bildschirm, dem Anwenderprogramm und der Systemsoft- und Hardware.

Zur Definition von Dialogen (=Bildern) wurde eine Dialogsteuersprache geschaffen, die nur aus Makroaufrufen besteht. Alle Aufrufe zur Steuerung des Dialoges in einem Bild bilden einen "Steuersatz".



Dieser wird durch ein zentrales Paket von Unterprogrammen interpretiert, das von den sichtgerätespezifischen Dialogprogrammen aufgerufen wird und seine Aufgaben zum Teil durch Unterprogramm-Aufrufe löst. Durch den Steuersatz können auch Anwenderprogramme (= Funktionsprogramme) beauftragt werden, je nach Definition des Dialoges. Daß heißt, der Dialogteil ist federführend, er delegiert an Programme die Aufgaben, die er nicht selbst erledigen kann bzw. will. Durch den eigenständigen Dialogteil ist die Trennung von Dialogdefinition und Funktionsprogrammen möglich. Dadurch bleiben Änderungen in der Dialogdefinition ohne Auswirkungen auf die Funktionsprogramme und durch die einmalige Erstellung von AUMADISY ergibt sich Zeitersparnis und eine erhöhte Fehlersicherheit.

3. Gründe für eine Eigenentwicklung

Sie sind durch die Nachteile der damals (Anfang '79) verfügbaren Standard-Software MASK 300 und SIDADIS gegeben.

Z.B. läßt SIDADIS keine segmentierten Masken zu, jedes Bild besteht aus nur einer Maske.

Hier die für uns wichtigsten Vorteile segmentierter Masken:

- jede Date hat nur eine Maske. Deshalb hat diese Date, die in vielen verschiedenen Bildern auftreten kann, immer dasselbe Erscheinungsbild. Bei Änderungen braucht also nur eine Maske geändert werden.
- oft vorkommende Dialogteile, z.B. eine 'Artikel-Anwahl' oder 'Tabellen-Ausgabe', können als externes Dialogteil vollständig, mit allen Maskenaufrufen und der dazugehörenden Bearbeitung, beschrieben und in den aktuellen Dialog "eingebunden" werden.
- dynamischer Aufbau eines Bildes. Dieser ist abhängig von den aktuellen Daten oder von Dateiinhalten.
- kleinerer Masken-Pufferbedarf, da eine Maske, die n-mal hintereinander aufgerufen wird, nur einmal Hauptspeicher Platz belegt.



MASK 300 bietet zwar die Möglichkeit, Bilder segmentiert aufzubauen, (daß heißt, jedes Bild besteht aus einer oder mehreren Masken) aber mit folgenden Nachteilen, die durch AUMADISY umgangen werden:

- die Akkumulatoren sind trotz mehrerer Masken nur einmal vorhanden
- die gezielte Bewegung des Cursors ist in Abhängigkeit von logischen Prüfungen nur innerhalb einer Maske (=Segment) möglich.
- bei logischen Prüfungen sind nur zwei Richtungen, abhängig von dem Ergebnis wahr oder falsch möglich. Es besteht keine Möglichkeit einer Mehrfachverzweigung.
- die Prüfergebnisse bleiben für den Bediener unsichtbar. Es wird nur der Cursor bewegt und es besteht keine Möglichkeit, Hinweis-texte auszugeben.
- es ist nicht möglich, in gleichen Masken, je nach der Logik des Bildes, unterschiedliche Prüfungen zu programmieren.
- das Attribut "Muß"-Feld ist zu starr. Es wird bei einem "Muß"-Feld auch dann das Überspringen abgelehnt, wenn das Feld durch eine Vorbesetzung ausgefüllt ist.

4. Sprachumfang

Die zur Dialogdefinition geschaffene Dialogsteuersprache besteht nur aus Makroaufrufen, die unter Ausnutzung der Möglichkeiten der Makrosprache MAS 300 selbst definiert wurden.

Für jede Aktion, z.B. Umspeichern einer Date, gibt es einen Aufruf, z.B. \oint UMSPEICH" ...;. Die Namen der Aufrufe wurden möglichst selbstdokumentierend gewählt. Ferner werden auch zur Angabe der aktuellen Parameter, z.B. Quelle und Ziel beim Umspeichern, nur sogenannte Kennwortparameter verwendet.

Tabelle 1: Übersicht der Aufrufe

Aufruf	Erläuterung
UMSPEICH	Feldweises Umspeichern, ggf. mit Konvertierung
UMSPTAFD	Umspeichern von Tabellen in Felder und
UMSPFDTA	umgekehrt, ggf. mit Konvertierung



SCHOTT

TRANSDATFD Daten von Dateien in Felder transferieren
TRANSFDDAT und umgekehrt
ARITFDFFD Rechnungen und logische Operationen zwischen Feldern
PROGBEAU Programme beauftragen (über \$KOOR"+)
VERGPRWE Vergleichen (auch Masken-Vergleiche) eines Feldes mit einem
VERGGRWE Feld, je einem Unter-, Obergrenzfeld
VERGTAWF und mit einem Tabellenfeld
SUCHTAWF Suchen mit beliebiger Bedingung in einer Tabelle
WAHLFORT Verzweigen mittels einer Index-Variablen
WIEDERHOLE Schleifenbildung mit log. Ausdruck als Abbruch-Bedingung
LOKBLOK Bildung von UP-ähnlichen Strukturen
RUFLOKBLOK
HOLEXTBLOK Einbindung von externen Steuersatz-Teilen
SEGMENT Aufruf einer Maske, Beschreibung der variablen Felder
BILDAUS Ausgabe des Bildes und Dialogbeginn
BILDABBR Abbruch des Dialoges in diesem Bild
CURSPOS Positionierung des Cursors auf beliebige Felder
SEFDOBE Beschreibung der dynamischen Bearbeitung vor oder nach einem
SEFDNABE Ein- oder Ausgabe-Feld einleiten
SEFDSOBE Beschreibung der Steuerzeichen Bearbeitung einleiten
FELDSIGAUS Ausgabe von Text-Feldern auf Sichtgerät
HINWXTAUS Ausgabe von Hinweistexten auf Sichtgerät
FIXTEXTAUS

-- Um den aktuellen Wert des Kennwortparameters anzugeben, werden in der Regel sogenannte "IDENTA's" verwendet. Das sind symbolische Adressen im Sinne von ASS 300, die mit IDENT-Anweisungen in zentralen Quellsprache-Elementen definiert sind und über die AKEI-Anweisungen mit übersetzt werden. Dadurch wird die fehlerhafte Adressierung einer Date weitgehend vermieden. Außerdem kann die Korrektheit der Angabe durch den Makroübersetzer geprüft werden, da die "IDENTA's" mit einer charakteristischen Zeichenfolge beginnen.

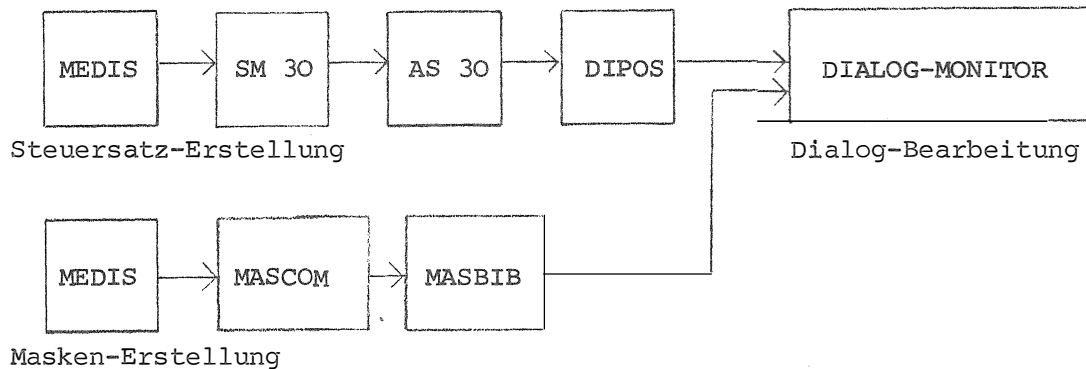


Bild 4: Erstellung eines Dialoges

Der "Steuersatz" zur Steuerung des Dialoges in einem Bild wird z.B. mit dem Editor MEDIS erzeugt. Er wird vom Makroübersetzer SM 30 vorübersetzt, der die syntaktische Korrektheit der Aufrufe und ihrer Parameter überprüft. Es wurde Wert darauf gelegt, während der Übersetzungszeit möglichst viele Prüfungen durchzuführen, sogar zu Lasten der Übersetzungszeit.

Danach erfolgt durch den Assembler AS 30 die Übersetzung in laadbare Grundsprache. Dabei werden die "IDENTA's" ausgewertet und fehlende oder falsche Definitionen erkannt.

Die Grundsprache wird in eine Datei geladen und von dort nach Bildanwahl vom Steuersatzinterpreter gelesen und abgearbeitet.

Parallel dazu müssen die benötigten Masken editiert, von MASCOM kompiliert und vom MASBIB geladen werden.

Während der Abarbeitung des Steuersatzes werden die logisch richtige Reihenfolge der Aufrufe und zulässigen Adressierungsgrenzen geprüft.

5. Datenbereiche und Adressierungsvarianten

Es existieren verschiedene Datenbereich, die getrennt adressiert werden:

- das Arbeitssatz-Feld als Schnittstelle zwischen Dialog und Funktionsprogrammen:

...ASAD @ ident, fdreladr

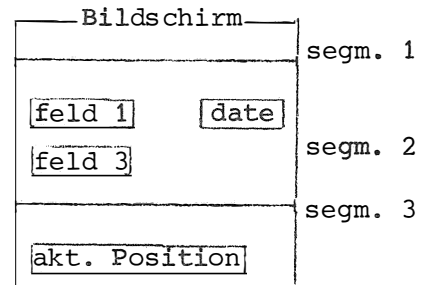
Arbeitssatz	
ident	
	fdreladr
date	
Arbeitss.-Feld 1	



SCHOTT

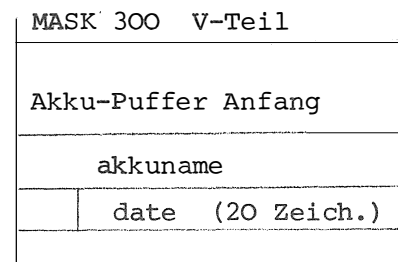
- der Nettodaten-Puffer als Schnittstelle zu MASK 300 zum Lesen der eingegebenen Daten oder zum Vorsetzen von Ein- und Ausgabefeldern vor der Bildausgabe:

...SEFD P -1,2,3



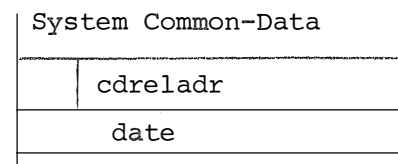
- die Akkumulatoren als Schnittstelle zwischen dem Dialog und MASK 300.
Z.B. zur Ausgabe von Daten während des Dialoges:

... AKNA P akkuname



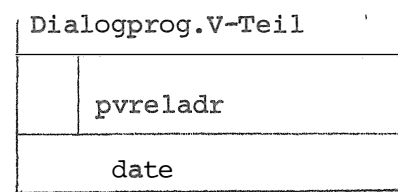
- der System-CD, zum Lesen von Systemkonstanten

... CDAD P cdreladr



- der Dialogprogramm-Variablenteil, zum Zugriff auf aktuelle dialogspezifische Daten und auf Hilfszellen

... PVAD P pvreladr



Außerdem existieren noch spezielle Adressierungsvarianten zur Angabe von Konstanten direkt im Aufruf oder zur Adressierung von Daten in Tabellen.

Allen Adressierungsvarianten ist gemeinsam, daß sie nur den Anfang einer beliebig langen Date bzw. Datenfolge bezeichnen. Deshalb wird auch von Feldern gesprochen. Die Größe dieser Date bzw. dieses Feldes geht entweder aus der zu dem Aufruf gehörenden Strukturbeschreibung hervor oder sie liegt eibdeutig fest. Die Strukturbeschreibung erfolgt mit einem Kennwortparameter ...STRU.

Bei allen Adressierungen wird nur eine Distanz-Adresse zum Anfang des jeweiligen Datenbereiches angegeben. Dadurch wird die Beschreibung eines Dialoges unabhängig von der Lage des AUMADISY im Haupt-



SCHOTT

speicher. Beispiel für einen Aktions-Aufruf: Es soll die Date X,
eine Zahl die ein Wort groß ist und im System-CD an Wort 24 steht,
an den Platz der Date Y, im Wort 4 des Variablen-Teils des Dialog-
Programmes umgespeichert werden: bzw. bei Benutzung der "IDENTA's"

§ UMSPEICH

"FELDSTRU P WZAHN 1
"QUELCDAD P 25
"ZIELPVAD P 4

;

§ UMSPEICH

"FELDSTRU P WZAHN 1
"QUELCDAD P CSDATE X
"ZIELPVAD P PVDAT Y

;

6. Beispiel Dialogsteuersatz

Es soll folgendes Bild ausgegeben werden, das aus zwei Segmenten
besteht:

Segment 1 AUSGABE-FELD 1 : EINGABE-FELD 2 : _____

Segment 2 EINGABE-FELD 1 : _____

Bildschirmrest ist unbenutzt (leer)

Der Steuersatz soll:

1. ein Datum in Segment 1, Feld 1 ausgeben
2. den Cursor auf Eingabefeld 2 von Segment 1 positionieren
 - 2.1 die Eingabe in ein Arbeitssatz-Feld umspeichern
 - 2.2 ein Programm beauftragen
 - 2.2.1 nach erfolgreichem Ende den Cursor auf Segment 2, Feld 1 stellen
 - 2.2.2 sonst bleibt er auf dem aktuellen Feld (Segment 1, Feld 2)
3. in Segment 2, Feld 3 wird geprüft, ob Eingabe=ABBR (=Abbruch)
 - 3.1 falls Ja, wird die (dynamische) Bildbearbeitung beendet
 - 3.2 falls Nein, springt der Cursor auf Segment 1, Feld 2.

--

Der Steuersatz in "symbolischer Schreibweise" (Kommentar in %...%):

§ SATZANF" ... ; %Einleitung des Steuersatzes %

% Masken Aufrufe: %

%	Segment-Nr.	Masken-Name	Zeilenanz.	%
---	-------------	-------------	------------	---

§ SEGMENT"	1	"MASKE P MASKE 1	"ZEILENZAHN P 2	%
------------	---	------------------	-----------------	---

%		Feld 1	Feld 2	%
---	--	--------	--------	---

; "Feldliste P ALV4 , ERN5



SCHOTT

```

§ BILDAUS; % Ausgabe aller bisher aufgerufenen Masken %
% ab hier: dynamische Bearbeitung, abhängig von den Eingaben bzw. %
%                               Cursorbewegung durch den Bediener %
% Vorbereitung für Segment 1, Feld 1 %
§ SEFDVOBE" 1"1 %
    "§UMSPEICH" DATUM      FELD 1      ;
;
%Nachbearbeitung für Segment 1, Feld 2 %
§ SEFDNABE" 1"2
    "§UMSPEICH" Eingabe      ASFD, DATE 1      ;
    "§CURSPOS"  SEGM 2 " FELD 3      ;
    % Funktionsprogramm xy beauftragen %
    "§PROGBEAU" PROGRAMM XY " KENNUNG XX " ERGEBNIS VARIABLE ;
    "§WAHLFORT" ERGEBNIS VARIABLE " ADRLIST 0 MISS ERF , ERF ;
§SYAD"MISS ERF;
    "§CURSPOS " SEGM 1 " FELD 2 ; % Sprung nur bei Fehler %
§SYAD"      ERF;
%Nachbearbeitung für Segment 2, Feld 1 %
§SEFDNABE" 2"1
    "§CURSPOS " SEGM 1 " FELD 2      ;
    % Abfrage: Bearbeitung abbrechen (Eingabe=ABBR)??? %
    "§ VERGPRWE" EINGABE " = " ABBR      " ERGEBNIS VARIABLE;
    "§ WAHLFORT" ERGEBNIS VARIABLE " ADRLIST 0 NO ABBR,ABBR;
§ SYAD"      ABBR;
    "§ BILDABBR;
§ SYAD"NO ABBR;
;
§ SATZEND" ...; % ende eines steuersatzes %

```

Tabelle 2: Angaben zur Steuersatz-Erstellung:

Steuersatz-Umfang	obiges Beispiel	klein	groß	
Quellsprache	90	370	1800	Zeilen
	27	53	345	Aufrufe
Grundsprache	125	700	1500	Worte



SCHOTT

Übersetzungszeiten SM 30	6	17	66	Minuten
in Minuten, AS 30	5	5	19	Minuten
incl. Protokoll-				
Ausgabe Summe	11	22	85	Minuten

7. Zusammenfassung

Mit AUMADISY wurde das "Lager-Platz- und -Bestands-Verwaltungssystem" erstellt. Dabei haben sich vor allem die strikte Trennung zwischen Dialogdefinition und problemspezifischem Funktionsprogramm und der modulare Aufbau des zentralen Dialogmonitors als vorteilhaft erwiesen. Dadurch konnten bei fortschreitender Realisierung des Projektes Aufrufe für spezielle Aktionen, deren Notwendigkeit bei Projektbeginn noch nicht abzusehen war, leicht geschaffen werden. Seit Fertigstellung im Herbst '81 wurde AUMADISY bei drei weiteren internen Anwendungen eingesetzt.