

A N D I :

Ein anwendungsunabhängiges Dialogsystem

A. Alberti, W. Geiger, V. Jarsch, G. Tretter

Kernforschungszentrum Karlsruhe
Institut für Datenverarbeitung in der Technik
D-7500 Karlsruhe, Postfach 3640

1. Einleitung

ANDI ist ein universelles System zur Definition (Programmierung) von rechnergeführten Dialogen. Es ist in Form eines überwiegend in Fortran geschriebenen Unterprogrammpaketes realisiert, dessen Komponenten im Anwenderprogramm aufgerufen werden /1/.

Entwickelt wurde ANDI im Rahmen der Erstellung des Kernmaterial-Informationssystems KANIS (Karlsruhe Nuclear Information System) am Institut für Datenverarbeitung in der Technik (IDT) des Kernforschungszentrums Karlsruhe (KfK). Dieses System dient den internationalen Atomenergiebehörden und den Betreibern kerntechnischer Anlagen zur Überwachung des Kernmaterialflusses. Dem Benutzer stellt sich das Informationssystem dar in Form von Dialogen für die Bestandsaufnahme, Prozeßdialogen und Komponenten für die Erstellung diverser Berichte. Als Rechner wird eine Siemens R30 eingesetzt, die Kommunikation erfolgt über Bildschirmeinheiten Siemens 3974.

Das Dialogsystem bildet das Interface zwischen Bediener, Anwenderprogramm und Systemsoft- und Hardware (Bild 1). Die Systemkomponenten beinhalten die für die realisierten Dialogfunktionen erforderliche Kontrollstruktur, und ermöglichen eine Dialogprogrammierung auf hohem Niveau.

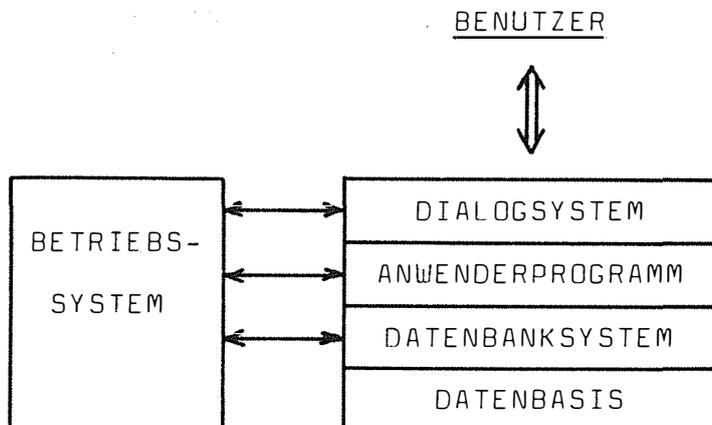


Bild 1: Struktur des Kernmaterial-Informationssystems

2. Warum eigenständiger Dialogteil

Dialogprogramme können auch mit Hilfe von z.B. FORTRAN Read/Write-Statements programmiert werden. Im folgenden sind einige der Gesichtspunkte aufgeführt, die die Entwicklung eines universellen Dialogsystems begründen.

A Erleichterung der Dateneingabe

Merkmale und Funktionen, die die Arbeit des Dialogführenden unterstützen:

- Anpassung an die dem Benutzer gewohnte Darstellungsweise und Terminologie (z.B. freier Bildaufbau, formatfreie Eingaben, führende Nullen und Blanks zugelassen).
- Bereitstellung leistungsfähiger Kommandos zur Dialogunterstützung.
- Komfortable Fehlerbehandlung:
Frühzeitige Fehlererkennung, detaillierte Meldung von Fehlerursache und -Art, das Angebot von auf die möglichen Fehlerarten zugeschnittenen Korrekturverfahren.

B Einfache Dialogdefinition

Der Programmierer wünscht sich:

- Dialogprogrammierung mit Hilfe weniger Subroutinen auf hohem Niveau.
- Einfache Programmstruktur.
- Verdeckte Kontrollstruktur:
Die Steuerungsmechanismen für Kommandoabläufe und Fehlerbehandlung sind in den Dialogteileroutinen enthalten.

C Trennung von Dialogdefinition und anwendungsspezifischer Software

- Änderungen in einer Dialogdefinition oder den Dialogsystemfunktionen bleiben ohne Auswirkung auf das Anwenderprogramm.
- Einmalige Erstellung des Dialogsystems.
Dadurch Zeitersparnis und erhöhte Fehlersicherheit.

3. Das Dialogsystem ANDI

Wie werden die Gesichtspunkte, die zur Entwicklung eines Dialogsystems führten, in dem realisierten System repräsentiert?

3.1. Die Dialogdefinition und der Dialogablauf

In Bild 2 ist die Grobstruktur eines Dialogprogrammes bei Anwendung der Subroutinen des Dialogsystems dargestellt.

Im Programmkopf stehen wie üblich die Vereinbarungen und Initialisierungen. Es folgt der Dialogrumpf, der die eigentliche Dialogdefinition enthält. Im Anschluß beginnt der problemspezifische Programmteil, in dem - bei Dialogen zur Datenerfassung - die eingelesenen Daten verarbeitet und, falls eingesetzt, mit Hilfe von Datenbankfunktionen in der Datenbasis abgelegt werden.

Der Dialogrumpf besteht aus Aufrufen von Routinen des Dialogsystems, mit denen die gewünschte Folge von Stringausgabe, Datenausgabe und Dateneingabe programmiert wird. Diese Dialogdefinition ist eingeschlossen in eine Initialisierungsroutine und eine den Dialog abschließende Funktion.

Vereinbarungen, Initialisierungen

```
<MARKE> call START (Parameter)
          •
          •
          •
          call PRINTS (Parameter)
          •
          call PRINT (Parameter)
          •
          call LIES (Parameter)
          •
          •
          •
          if (.not. ENDE (Param.) go to <MARKE>
```

Problemspezifischer Programmteil

Bild 2: Grobstruktur eines Dialogprogrammes mit ANDI-Routinen

Im wesentlichen erfüllen die einzelnen Dialogsystemroutinen folgende Aufgaben:

- START** Diese Routine steht zu Beginn einer Dialogdefinition und beinhaltet die Initialisierung von Parametern des Dialogsystems. Außerdem wird der Bildschirm des Sichtgerätes gelöscht.
- PRINTS** Routine zur Ausgabe von Strings.
- PRINT** Datenausgaberoutine.
- LIES** Dateneingaberoutine. Sie beinhaltet die Ausgabe eines Vorschlagswertes, das Einlesen eines Datums, die Behandlung von Eingabefehlern, die Erhöhung des Dialogschrittzählers und den Anstoß aufgerufener Dialog-Kommandos.
- ENDE** Die Dialogdefinition abschließende Funktion. Sie dient der Behandlung von im Dialogrumpf nicht zu behebenden Fehlern und zur Ablaufsteuerung von Dialog-Kommandos.

Je nach Verlauf des vorangegangenen Dialoges wird hier die Frage 'DATEN IN ORDNUNG?' oder 'DATEN FEHLERHAFT. ABBRECHEN?' ausgegeben. Bei Beantwortung mit 'JA' wird das Programm mit dem problemspezifischen Teil fortgesetzt, d.h. der Dialogrumpf verlassen. Im zweiten Fall ('DATEN FEHLERHAFT...') wird hierbei eine entsprechende Fehlererkennung übergeben. Werden die Fragen mit 'NEIN' beantwortet, beginnt der Dialog erneut (Sprung zu der mit <MARKE> versehenen Routine START). In diesem Falle werden die zuvor eingegebenen Daten als Vorschlagswerte ausgegeben. Sie können durch Drücken der Eingabetaste akzeptiert oder überschrieben werden.

Der normale Dialogablauf ist in Bild 3 graphisch dargestellt.

Zum System gehören weiter Routinen zur Fehlermeldung und -Registrierung (SFEHL), für Seitenwechsel bei mehrseitigen Dialogen (SEITWE), für die Ausgabe einer Hardcopy des Dialoges (HCALLE), usw.

3.2. Die Parameter der Dialogsystemroutinen

Die bei Aufruf der Routinen des Dialogsystems zu übergebenden Parameter spiegeln das Niveau der Programmierung wieder. Als Beispiel sei die Dateneingaberoutine LIES betrachtet, die mit folgenden Parametern versehen ist:

- USR Ein Informationsblock, auf den alle Dialogteileroutinen zugreifen, und der z. Zt. 25 Einzelinformationen enthält. Dazu gehören z. B. Programmzustand, Dialogschrittzähler, logische Gerätenummern der Ein- und Ausgabegeräte, usw.
- TRAFO Der Name der Transformations- und Prüfroutine für das einzulesende Datum. Sie wandelt das Eingabedatum von der externen in die interne Darstellung um und trägt es in einen Datensatz ein. In dieser Routine können Syntax- und Plausibilitätsprüfungen, Existenzprüfungen und Prüfungen auf Zulässigkeit der Eingabe bei korrelierten Daten vorgenommen werden.
- XPOSIT X- und Y-Position, ab der auf dem Bildschirm Eingaben eingelesen werden.
- YPOSIT
- TEXTUR Darstellungsart auf dem Bildschirm (normal, invers, blinkend).
- LGA Formatangaben für numerische Werte.
- DEZI
- sieben Daten-sätze Vorgesehen zur Übergabe der Daten, die Gegenstand des Dialoges sind.
- TPPART Die Bezeichnung des Datensatzes, auf den sich die jeweilige Eingabe bezieht.
- ATTNR Die Nummer des Attributes im Datensatz, das dem Eingabedatum zugeordnet ist.

Die Datenausgaberoutine PRINT und die Dateneingaberoutine LIES besitzen die größte Anzahl von Parametern. Die Routine zur Stringausgabe benötigt nur sechs Parameter; die Start-, die Abschluß- und die Mehrzahl der weiteren zur Verfügung stehenden Dialogsystemroutinen benötigen lediglich den Informationsblock USR.

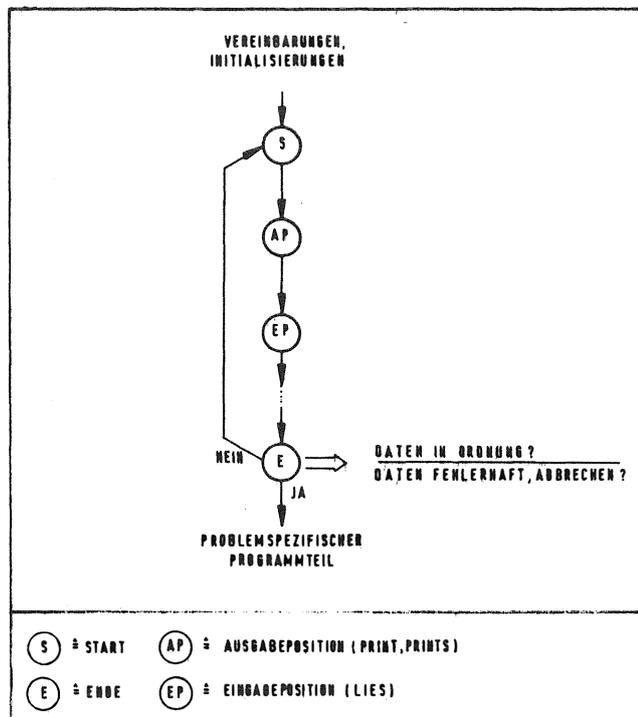


Bild 3: Normaler Dialogablauf

3.3. Die Kommandos

Als Hilfen für die Dialogführung stehen mehrere Kommandos zur Verfügung. Ihr Aufruf ist an jeder Eingabeposition des Dialogrumpfes möglich. Eine an der Eingabeposition anstehende Eingabeanforderung geht nicht verloren; das gleiche gilt für einen vorhandenen Vorschlagswert.

Die den Dialogablauf beeinflussenden Kommandos sind SCHNELLDURCHLAUF, KORRIGIEREN und ABBRUCH. Weitere Kommandos dienen der Ausgabe einer HARDCOPY des Sichtgeräteebildes, sowie zur Neuwahl der Ausgabegeräte für Hardcopy bzw. Testausgaben. Die drei erstgenannten Kommandos haben Einfluß auf den Dialogablauf (Bild 4):

Das Kommando SCHNELLDURCHLAUF dient bei mehrseitigen Dialogen dazu, schon abgearbeitete Dialogseiten erneut auf dem Bildschirm des Sichtgerätes darzustellen. Nach Aufruf wird ab Dialogbeginn jeweils eine Seite ausgegeben; erst nach Betätigung der Eingabetaste wird mit der Ausgabe der folgenden Seite begonnen. Ist die Eingabeposition des Kommandos erreicht, wird der Dialog normal fortgesetzt.

Das Kommando KORRIGIEREN dient zur Korrektur bereits eingegebener Daten. Nach Anstoß des Kommandos werden ab Dialogbeginn die zuvor eingegebenen Daten als Vorschlagswerte ausgegeben. Sie können durch Drücken der Eingabetaste bestätigt oder durch neue Werte überschrieben werden. Ist die Eingabeposition des Kommandos erreicht, wird der Dialog normal fortgesetzt.

Mit dem Kommando ABBRUCH kann ein laufender Dialog vorzeitig beendet werden. Der Dialogrumpf wird bei Ausführung dieses Kommandos verlassen. Im problemspezifischen Programmteil kann diese Aktion an dem im Informationsblock USR gelieferten Programmzustand erkannt werden.

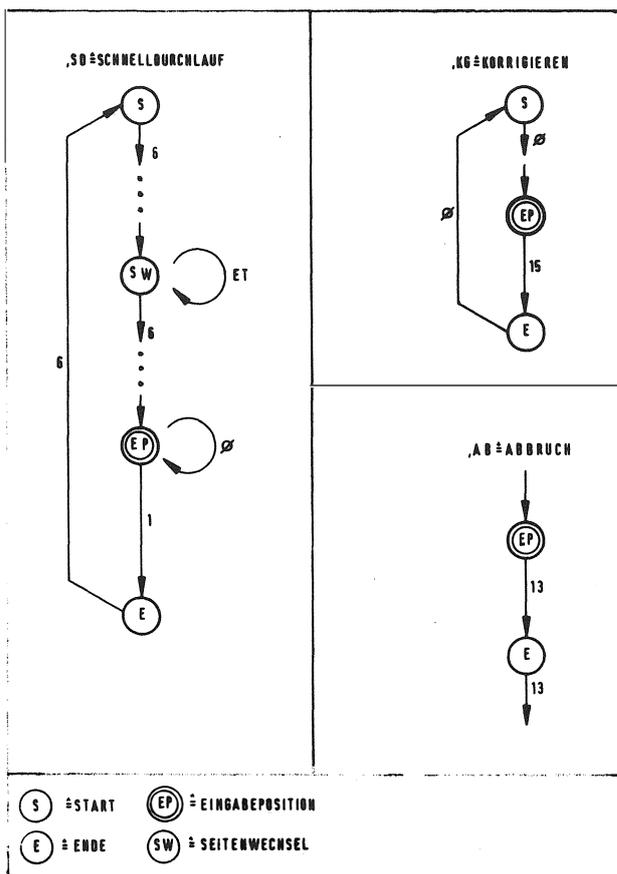


Bild 4: Ablauf der Kommandos SCHNELLDURCHLAUF, KORRIGIEREN und ABBRUCH.

3.4. Fehlerdiagnostik und Fehlerbehandlung

Prinzip ist, Fehler zum frühestmöglichen Zeitpunkt zu erkennen und ihrer Ursache entsprechend zu reagieren. Dies dient drei Zielen:

- Unterstützung des Dialogführenden.
Ihm bleiben Korrekturen größeren Umfangs damit weitgehend erspart.
- Sicherung der Zuverlässigkeit der Daten.
Verhindert werden soll, daß falsche Daten in die Datenbasis gelangen.
- Unterstützung des Programmtests.
Programmfehler sind durch die Vielzahl der Prüfungen und detaillierte Fehlermeldungen schnell einzukreisen.

Die Meldungen erfolgen je nach Fehlerklasse - es wird zwischen sechs Klassen unterschieden - in der letzten Zeile des Sichtgerätes, auf der Systemkonsole oder auf beiden Geräten. Ausgegeben werden Fehlercode, Fehlerklasse und Benutzeridentifizierung. Zur Zeit wird vorbereitet, statt des Fehlercodes die komplette Erläuterung auszugeben.

Der weitere Dialogablauf nach der Erkennung eines Fehlers hängt ebenfalls von der Fehlerklasse ab. Bei den korrigierbaren Eingabefehlern erfolgen bis zu zwei weitere Eingabeanforderungen. Nicht korrigierbare Eingabefehler führen zu einem Abbruch des Dialoges und zu der Abfrage, ob eine Wiederholung gewünscht wird (In der Dialog-Abschlußroutine ENDE). Die sogenannten Systemfehler haben ebenfalls einen Abbruch des Dialoges zur Folge, der Dialogrumpf wird in diesem Fall jedoch verlassen, d.h. es erfolgt keine automatische Abfrage auf Wiederholung.

4. Schlußbetrachtung

Auf der Basis des Dialogsystems ANDI wurden am IDT die Dialoge für das Kernmaterial-Informationssystem KANIS implementiert. Dabei haben sich vor allem die strikte Trennung von Dialogdefinition und problemspezifischem Programmteil und die Tatsache, daß der Dialogprogrammierer keine eigene Kontrollstruktur aufzubauen braucht, als sehr vorteilhaft erwiesen.

Der Programmtest wurde durch die leistungsfähige Fehlerbehandlung stark unterstützt.

Die Erfahrungen beim Anwender zeigen, daß auch in der Arbeit an DV-Anlagen ungeübte Benutzer mit den Funktionen und der Arbeitsweise des Dialogsystems schnell vertraut sind.

Referenzen:

- /1/ Jarsch, V.: Benutzerinterface für das SpFK-DV-System.
Kernforschungszentrum Karlsruhe, unveröffentlichter Bericht, 1979.
- /2/ Jarsch, V.; Landmark, K.; Müller, W.: Grundlagen und Entwurfsmethodik eines Systems für die Mensch-Rechner Kommunikation.
Kernforschungszentrum Karlsruhe, KfK-Bericht 2556, Januar 1978.