

Eine allgemeine Dialogschnittstelle in PEARL für die Kommunikation mit PEARL-Prozessen über Bildschirm und Tastatur

Dipl.-Math. Günter Stöhr

Kurzfassung

Im Rahmen von Betriebsleitsystemen für den öffentlichen Personennahverkehr hat die Kommunikation des Fahrdienstleiters mit dem Prozeß eine übergeordnete Bedeutung. Deshalb wurde bei der Entwicklung des vom BMFT geförderten rechnergesteuerten Betriebsleitsystems BON auf die Realisierung eines allgemein verwendbaren Dialogkonzepts mit einem zentralen Dialogprogramm besonderer Wert gelegt.

Die Lösung der Aufgabe wurde in Form eines PEARL-Programms erreicht, das zu Dialog-Anwendungsprogrammen dazugebunden werden kann. Die Schnittstellen stehen als Unterprogrammaufrufe zur Verfügung. Die Ausgabeseite ist in Form einer sog. Einheitlichen Grafik-Schnittstelle (EGS) für Bildschirmgeräte verwirklicht, so daß bei Prozeßanwendungen übliche semigrafische Bildschirmgeräte aber auch alphanumerische Terminals verwendet werden können.

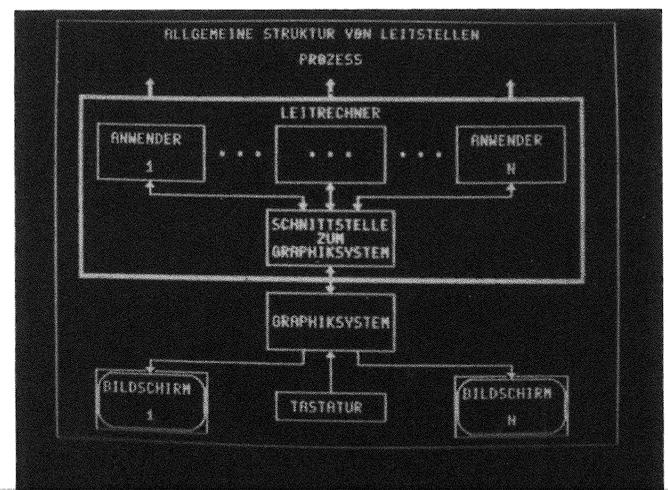
Abstract

Within the scope of a computerised management system for public transport the communication of the operator with the process has a superior importance. Therefore in the face of developing the computerised management system BON in the town of Hannover, supported by BMFT - the German Ministry for Research and Technology -, there was set a high priority on the creation of a universal dialogue-module with a central dialogue-program.

available as procedure-calls. The output-interface of the dialogue-module is a so-called "Einheitliche Grafik-Schnittstelle" (Standardized Graphic-Interface), giving the possibility of using semigraphic screens as well as alpha-displays as peripheral devices.

1. Idee und Entwicklung der Dialog-Schnittstelle im Rahmen BON

Eine der wichtigsten Aufgaben bei der Entwicklung rechnergestützter Leitsysteme ist die Verbindung des Prozesses mit dem Bedienungspersonal. Darunter fallen Informationen über Betriebszustände, Prozeßbeeinflussung durch Steuereingaben, Anzeige kritischer Ereignisse usw. Betrachtet man diese Mensch-Prozeß-Schnittstelle bei verschiedenen Leit- und Überwachungssystemen genauer, so fällt immer wieder die Ähnlichkeit der Struktur der Leitstellen bezüglich Hard- und Software auf (siehe auch Bild 1):



The solution of the problems was achieved by a PEARL-program, which can be added to the dialogue-user-programs. The interfaces are

Bild 1: Allgemeine Struktur von Leitstellen

- Es gibt ein Ein- und Ausgabe-(Grafik-)System mit Bildschirm, Tastatur und ggfs. weiteren peripheren Einheiten
- Es gibt einen oder mehrere Leitrechner mit Programm-Modulen, die eine Verbindung zum Bedienungspersonal am Leitstellenarbeitsplatz benötigen
- Die Verbindung zwischen Leitrechner und Grafiksyste[m] wird über besondere Software-schnittstellen hergestellt
- ein zentrales Dialogprogramm mit einem allgemeinen Dialogkonzept entwickelt wurde, über das die Anwender-Module freizügige Dialoge mit dem Rechner führen können,
- zur Ansprache des Grafiksyste[m] eine "Einheitliche Grafik-Schnittstelle für die Anbindung semigrafischer Farbbildschirmsysteme" (im folgenden EGS) formuliert und angewendet wurde,
- die Realisierung der Komponenten in PEARL erfolgte, um nicht auf einen bestimmten Prozeßrechnerhersteller von vornherein festgelegt zu sein.

Darüber hinaus zeigt sich deutlich, daß die Formalismen der Dialogführung im Konzept oft identisch sind und sich nur in Ausprägungen unterscheiden.

Trotz dieser Gegebenheiten ist die softwaretechnische Realisierung sehr unterschiedlich. Dialog- und Schnittstellenbausteine werden immer wieder "neu erfunden". Gleichzeitig werden sie mit der eigentlichen Prozeßsoftware so verwoben, daß eine Übertragung auf andere Leitsysteme selbst bei quasi identischer Aufgabenstellung nicht mehr möglich ist.

Bei der Entwicklung des vom BMFT geförderten rechnergesteuerten Betriebsleitsystem für den öffentlichen Personennahverkehr (BON) wurde gemäß der Forderung der Standardisierung und der Entwicklungsfähigkeit diesen Erkenntnissen Rechnung getragen (siehe auch Bild 2), indem:

Dadurch sollten die Anwenderpakete (hier Disposition, Soll-Vorgaben, Statistik, Fahrgastinformation) weitgehend von der Führung der Dialoge entlastet werden sowie die einheitliche Bedienerführung sichergestellt werden. Weiterhin wird durch die Verwendung der EGS die Unabhängigkeit von einer speziellen Grafik-Hardware erreicht.

Im Rahmen BON wurde das Dialogprogramm dabei im sog. Modulpaket Betriebsinformation angesiedelt. Es ist aber an sich unabhängig. Für die Pilotanwendung in Hannover findet als Grafiksyste[m] ein Prozeß-Video-System PVS 1300 und als Leitrechner ein EPR 1500 der Firma Krupp-Atlas-Elektronik, Bremen (im folgenden KAE), Verwendung.

Im folgenden werden zunächst auf die Anforderungen an den Dialogbaustein und die sog. Einheitliche Grafikschnittstelle näher eingegangen. Anschließend wird die Realisierung in PEARL erläutert.

2. Anforderungen an den Dialogbaustein

Der Dialogbaustein soll von verschiedenen Bearbeitungsmodulen des BON-Systems - nämlich Soll-Vorgaben, Disposition, Statistik und Fahrgastinformation -, die eine Verbindung zum Disponenten, d.h. Bediener, benötigen, individuell benutzt werden können. Insgesamt wurden folgende Forderungen aufgestellt: (siehe auch Bild 3)

- freie Definition von Dialogformularen
- Eingabe beliebiger Werte in Eingabefelder
- Möglichkeit der Vorbelegung von Eingabefeldern mit Werten, die vom Disponenten akzeptiert oder korrigiert (überschrieben) werden können

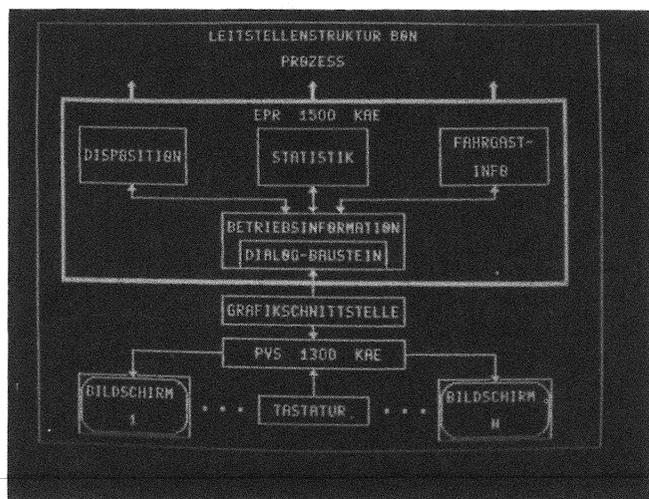


Bild 2: Leitstellenstruktur in BON

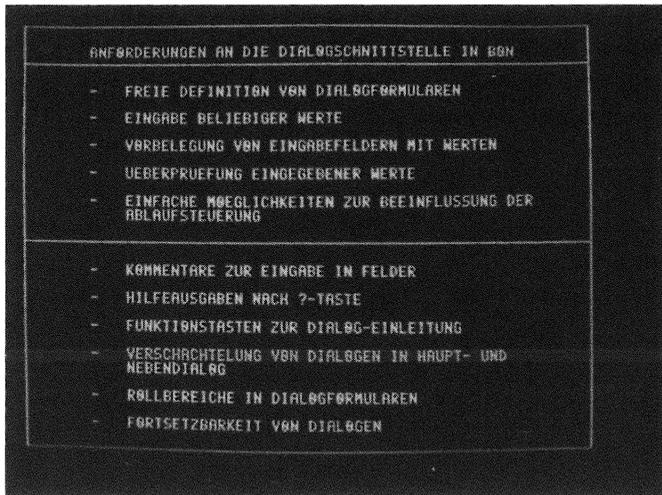


Bild 3: Anforderungen an die Dialog-Schnittstelle

- Überprüfung von eingegebenen Werten auf
 - * Erfüllung von Eingabekonventionen (Format-Prüfung)
 - * Sinnfälligkeit (Semantik-Prüfung)
 und, falls eine Fehleingabe erkannt wird, Ausgabe vorformulierter Fehlermeldungen mit anschließender Möglichkeit zur Korrektur der Eingabe
- einfache Möglichkeiten zur Beeinflussung der Ablaufsteuerung durch das anwendende Modulpaket

Ein Dialogformular besteht dabei im wesentlichen aus einem Grundbild (Raster) mit sogenannten Eingabefeldern, in die Werte sowohl ein- als auch ausgegeben werden können. Während einer Dialogführung gibt es immer genau ein aktuelles Eingabefeld, welches durch den Cursor (Schreibmarke) identifiziert wird. In dieses aktuelle Feld können Werte eingetragen oder, falls vorhanden, überschrieben oder akzeptiert werden. Jede abgeschlossene Eingabe in ein Feld wird überprüft und im Fehlerfall eine entsprechende Meldung ausgegeben. Anschließend wird dasselbe Eingabefeld zur Korrektur aktuell oder es wird, falls vorhanden, ein weiteres Feld behandelt. Ist kein weiteres Feld mehr vorhanden, so gilt der Dialog als abgeschlossen. Die Bedienführung in Form des Dialogablaufs kann dabei vom anwendenden Modulpaket leicht beeinflusst werden.

Zur Erhöhung des Komforts der Schnittstelle sowohl für den Bediener als auch für das anwendende Modulpaket wurden zusätzliche Anforderungen entwickelt:

- Ausgabe von Kommentaren vor der Eingabe zur Erläuterung des einzugebenden Wertes
- Ausgabe zusätzlicher, erläuternder Informationen, bezogen auf das aktuelle Eingabefeld, nach Betätigung einer Hilfstaste (?-Taste)
- Einsatz von Funktionstasten zur Dialog-Einleitung
- Verschachtelung von Dialogen in Haupt- und Nebendialoge
- Einführung von Rollbereichen in Dialogformularen
- Fortsetzbarkeit von Dialogen

Dies sind die wichtigsten Anforderungen, die im Rahmen der Entwicklung vor allem von der Anwenderseite aus zusammengetragen wurden. Desweiteren wurde von dieser Seite auch eine möglichst hardware-unabhängige Ansprache des Graphik-Systems gefordert. Darauf wird im nächsten Kapitel genauer eingegangen.

3. Die Grafikschnittstelle

Eine Anwenderforderung bei der Ansprache des Graphiksystems war die Hardware-Unabhängigkeit. Bei der Analyse verschiedener Graphiksysteme wurde nun festgestellt, daß diese einen im Prinzip ähnlichen logischen Aufbau besitzen. Dies legte die Schaffung einer Einheitlichen Grafik-Schnittstelle (EGS) nahe. Um eine eindeutige Trennung zwischen Leit-rechner und Grafiksystem zu erzielen, wurde von der Anwenderseite gefordert, Hard- und Softwareschnittstellen an derselben Stelle anzuordnen. Deshalb wurde für die zu entwickelnde EGS die Form einer Alphanumeric gewählt, die mit "PUT" und "GET" angesprochen werden kann. Als Graphiksystem werden dabei ein oder mehrere Graphikrechner mit Bildschirmen und Tastaturen, Lichtstift zugrunde gelegt. Als logische Grundeinheit wurden Bilder definiert, die geladen, auf Bildschirmen dargestellt und manipuliert werden können, indem etwa Texte eingetragen oder Farb- und Blinkinformationen hinzugefügt werden.

Zu einem Grafikbild gehören dann: (siehe auch Bild 4)

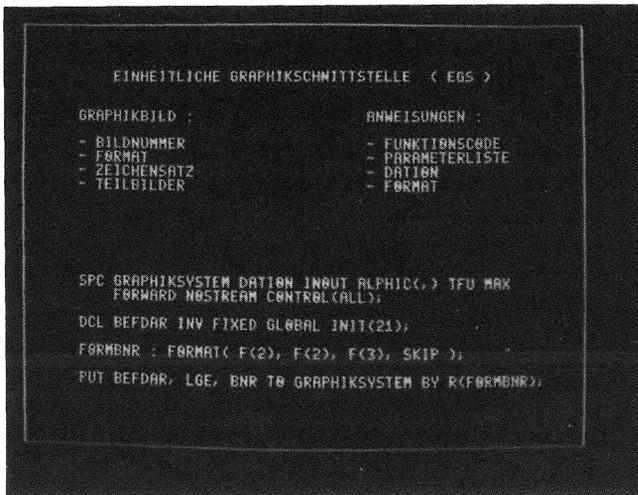


Bild 4: Die Einheitliche-Grafik-Schnittstelle

- eine Bildnummer
- ein Format (Zeilen * Spalten)
- ein Zeichensatz, mit dem das Bild erstellt wurde
- Teilbilder in verschiedensten Ausprägungen

Unter Teilbildern sind dabei etwa Bildauschnitte, vorgefertigte Großsymbole, Bildpositionen oder auch Eingabefelder zu verstehen. In einem Befehl an das Graphiksystem werden nun das anzusprechende Bild und die betroffenen Teilbilder usw. in einer Parameterliste zusammengefaßt. Der Befehl selbst enthält dann insgesamt:

- einen Funktionscode (BEF...)
- eine Parameterliste
- eine Dation
- ein Format

z.B. hat dann der Befehl an das Graphiksystem zum Darstellen eines Bildes mit der Nummer BNR auf dem Bildschirm, der im System unter dem logischen Namen LGE bekannt ist, das Aussehen:

```
PUT BEFDAR, LGE, BNR TO GRAPHIKSYSTEM
    BY R(FORMBNR);
```

Dabei ist BEFDAR der Befehlscode für Bild-Darstellen und FORMBNR das zugehörige Format.
 FORMBNR: FORMAT(F(2), F(2), F(3), SKIP);

In der hier angesprochenen Graphikschnittstelle werden dabei alle Funktionscodes, Formate und sonstigen grafikbezogenen Werte wie etwa Farben vom Implementierer des Gra-

fiksystems als benannte Konstanten zur Verfügung gestellt. Der Anwenderprogrammierer benötigt also keine speziellen Kenntnisse über Wertebereiche etc.

Im Falle der Entwicklung des Betriebsleitsystems BON ist der Dialogbaustein der Hauptanwender der EGS. Auf den Aufbau dieses Bausteins soll im folgenden eingegangen werden.

4. Aspekte der Realisierung des Dialogbausteins in PEARL

Die Dialogschnittstelle steht, von den sie benutzenden Modulpaketen aus gesehen, als Prozeduraufrufe zur Verfügung. Der Aufruf zur Anforderung einer Dialogführung sieht dabei wie folgt aus:

```
FUEHREDIALOG: PROCEDURE
```

```
(MP FIXED,          /* Auftraggeber          */
 KENNUNG FIXED, /* Kennnummer der
                   Aufforderung          */
 MNR FIXED,        /* Monitor              */
 BNR FIXED,        /* Bildnr. des Eingabe-
                   formulars          */
 FORTSETZUNG BIT(1), /* Dialog-
                   Fortsetzung?          */
 ANZEAFFELDER FIXED, /* Anzahl Ein-/
                   Ausgabefelder          */
 FELDEINTRAG() EINTRAG IDENT, /* Liste der
                   E/A-
                   Felder          */
 ERR FIXED IDENT) /* Return-Code          */
RESIDENT GLOBAL;
```

Der Auftraggeber identifiziert sich dabei gegenüber der Dialogschnittstelle durch seinen Namen sowie eine Kennnummer der Anforderung, die bei Dialogfortsetzung zur Feststellung der Zulässigkeit dient. Ferner wird die Bildnummer des Eingabeformulars, welches behandelt werden soll, angegeben. Der Dialogbaustein führt dabei unter dieser Nummer eine Beschreibung des Eingabeformulars, die alle wichtigen Informationen enthält. Dazu gehört unter anderem auch eine Beschreibung aller im Bild enthaltenen Ein-/Ausgabefelder mit: (siehe auch Bild 5)

- Anzahl möglicher Zeichen im Feld
- Liste der Nachbarteilbilder bei Betätigung einer der vier Cursor-Steuertasten
- zugehörige Konvertierungsroutine bei Ein-

```

/* DATENSTRUKTUREN DES DIALOG-BAUSTEINS */
TYPE EFELDER STRUCT /* E/A-FELD-BESCHREIBUNG */
  INZ      FIXED, /* ANZAHL MOEGL. ZEICHEN IM FELD */
  TAB      FIXED, /* NACHBARFELD NACH TAB → */
  TABNA    FIXED, /* NACHBARFELD NACH TAB ← */
  PFEILABEN FIXED, /* NACHBARFELD NACH TAB ↑ */
  PFEILUNTEN FIXED, /* NACHBARFELD NACH TAB ↓ */
  KONVROUT FIXED, /* KONVERTIERUNGSROUTINE */
  PRUEFROUT FIXED, /* PRUEFRROUTINE */
  ARBEITSIX FIXED, /* VERWEIS AUF ARBEITSPARAMETER */
  HELPLISTE FIXED, /* NAME DER HELPLISTE */

TYPE EINTRAG STRUCT /* EINTRAG IN EIN E/A-FELD */
  ITNR     FIXED, /* TEILBILDNUMMER */
  WERT     PARAMETER, /* EINZUTRAGENDER WERT */
  STATUS   FIXED, /* FELDSTATUS */
  MELDUNG  FIXED, /* KOMMENTAR ZUM FELD */

DER DATENTYP PARAMETER WIRD DABEI JE NACH ANWENDUNGSFALL
GESCHAFFEN.

```

Bild 5: Datenstrukturen der Dialog-Schnittstelle

- oder Ausgabe in das Feld
- zugehörige Prüfroutine bei Eingabe in das Feld
- Arbeitsparameter für die Konvertierungs- oder Prüfroutine
- Informationen über die Helpliste, die bei Betätigung der ?-Taste aufgeschaltet werden soll

Der Auftraggeber wählt nun aus allen möglichen Ein-/Ausgabefeldern im Formular diejenigen aus, die er im nächsten Dialog behandeln will, und liefert:

- ihre Anzahl und
 - ihre Feldeinträge
- an.

Ein Feldeintrag enthält dabei:

- die Teilbildnummer des Ein-/Ausgabefelds,
- den einzutragenden Wert,
- den Feldstatus bei Dialoganfang,
- den Kommentar, der bei Freigabe dieses Feldes ausgegeben werden soll.

Welches Aussehen der einzutragende Wert dabei haben kann, ist dem Anwendungsfall überlassen. Für das Forschungsvorhaben BON wurden zum Beispiel

- Zahlen
- Linien/Kurs-Nummer
- Haltestellenkürzel
- Datums-Angabe
- Abfahrzeiten

als Parameterarten benötigt.

Der Feldstatus ist die wichtigste Einflußmöglichkeit des Anwenders; er kann folgende Werte annehmen:

- Vor Eingabe:

Das Feld ist leer, und es soll eine Eingabe dort stattfinden.

- Vorbelegt:

Das Feld ist mit einem Wert vorbelegt, dieser muß akzeptiert oder überschrieben werden.

- Falsch:

Das Feld enthält eine falsche Eingabe.

- Veränderbar:

Das Feld enthält einen Wert, der verändert werden kann, aber nicht muß.

- Inaktiv:

Das Feld wird zur Zeit nicht benutzt.

- Ausgabefeld:

Das Feld hat keine Ausgabefunktion. Dort kann zur Zeit nichts eingegeben werden.

Mit Hilfe dieses Feldstatus ist somit eine einfache Einflußnahme auf den Dialogablauf möglich.

Die angegebenen Datenstrukturen sind als PEARL-Structs implementiert. Dabei wurde auf das TYPE-Konzept zurückgegriffen. Insbesondere ist dadurch auch die einfache Anpassbarkeit an den speziellen Anwendungsfall sichergestellt. Diese leichte Anpassbarkeit wird erreicht durch:

- die Arten der zu behandelnden Parameter sind vom Anwender vorgebar,
- die Konvertierungsroutinen zur Ein- und Ausgabe der Werte sind vom Anwender vorgebar,
- die Prüfroutinen zur Überprüfung der eingegebenen Werte sind vom Anwender vorgebar.
- bei der Software-Entwicklung wurden moderne Methoden aus der strukturierten Programmierung angewandt.

So sind zum Beispiel alle Datenbestände des Dialogbausteins in einem PEARL-Module zusammen mit ihren Zugriffsfunktionen zusammengefaßt. Dies bedeutet, daß die Daten nur über fest definierte Zugriffe erreichbar sind. Als Beispiel sei eine Procedure angeführt,

die die Cursorsteuerung in einem Eingabeformular übernimmt:

NXTFNRCUR: PROCEDURE

```
(MNR FIXED,          /* (E): Monitor */
BNR FIXED,          /* (E): Bildnummer*/
AKTCUR FIXED,      /* (E): Aktuelle
                  Cursorposition */
RICHTUNG FIXED,    /* (E): Richtung */
NEWCUR FIXED IDENT) /* (A): Neue
                  Cursorposition */
RESIDENT GLOBAL;
```

Diese Procedure bestimmt in einem Eingabeformular, daß durch seine Bildnummer identifiziert wird, das nächste Eingabefeld. Dazu benötigt die Procedure außer der alten Cursorposition und der Suchrichtung auch die Angabe des Monitors, auf dem das Eingabeformular dargestellt ist, da die Cursorsteuerung durch aktuelle Dialogwerte wie Feldstatus beeinflusst wird. Diese aktuellen Dialogwerte sind nun monitorbezogen im Datenbereich hinterlegt.

Alle wichtigen Dimensionierungen wie Anzahl Monitore, Anzahl von Eingabefeldern, maximale Anzahl von Ein-/Ausgabefeldern sind generierbar. Dies bedeutet, daß, je nach Anwendungsfall, auf einfache Weise durch ein PEARL-Programm eine optimal an die Erfordernisse angepaßte Version des Dialogbausteins erstellt wird.

Implementiert wurde der Dialogbaustein auf einem Rechner vom Typ EPR 1400 der Firma KAE in 3 PEARL-Modules:

- Datenbereich mit Zugriffsfunktionen (40 Prozeduren)
- Dialogbearbeitung (1 Task, 8 Prozeduren)
- Prüf- und Konvertierungsroutinen

Der Speicherplatzbedarf des im Anwendungsfall stark dialogorientierten Betriebsleitsystems BON beträgt mit Daten etwa 30-40 K-Worte. Für kleinere Anwendungen reduziert sich der Speicherbedarf entsprechend.

5. Schlußbemerkung

Die im Rahmen der Entwicklung des rechnergestützten Betriebsleitsystems geforderte Standardisierung und die hohen Anwenderanforderungen bezüglich Komfort und Geschwindigkeit, machten die Entwicklung des Dialogbausteins sowie einer allgemeinen Grafikschnittstelle notwendig. Zusammenfassend kann man nach eingehenden Tests des Dialogbausteins mit mehreren Leittischen sagen, daß dieser die Anforderungen erfüllt. Es besteht die begründete Aussicht, daß er sich auch in der betrieblichen Erprobung bewährt.

Anschrift des Autors:

Dipl.-Math. Günter Stöhr
Büro
Dr.-Ing. H. Heusch/
Dipl.-Ing. J. Boesefeldt GmbH
Liebigstraße 20

5100 Aachen

Tel.: 0241/16 50 01