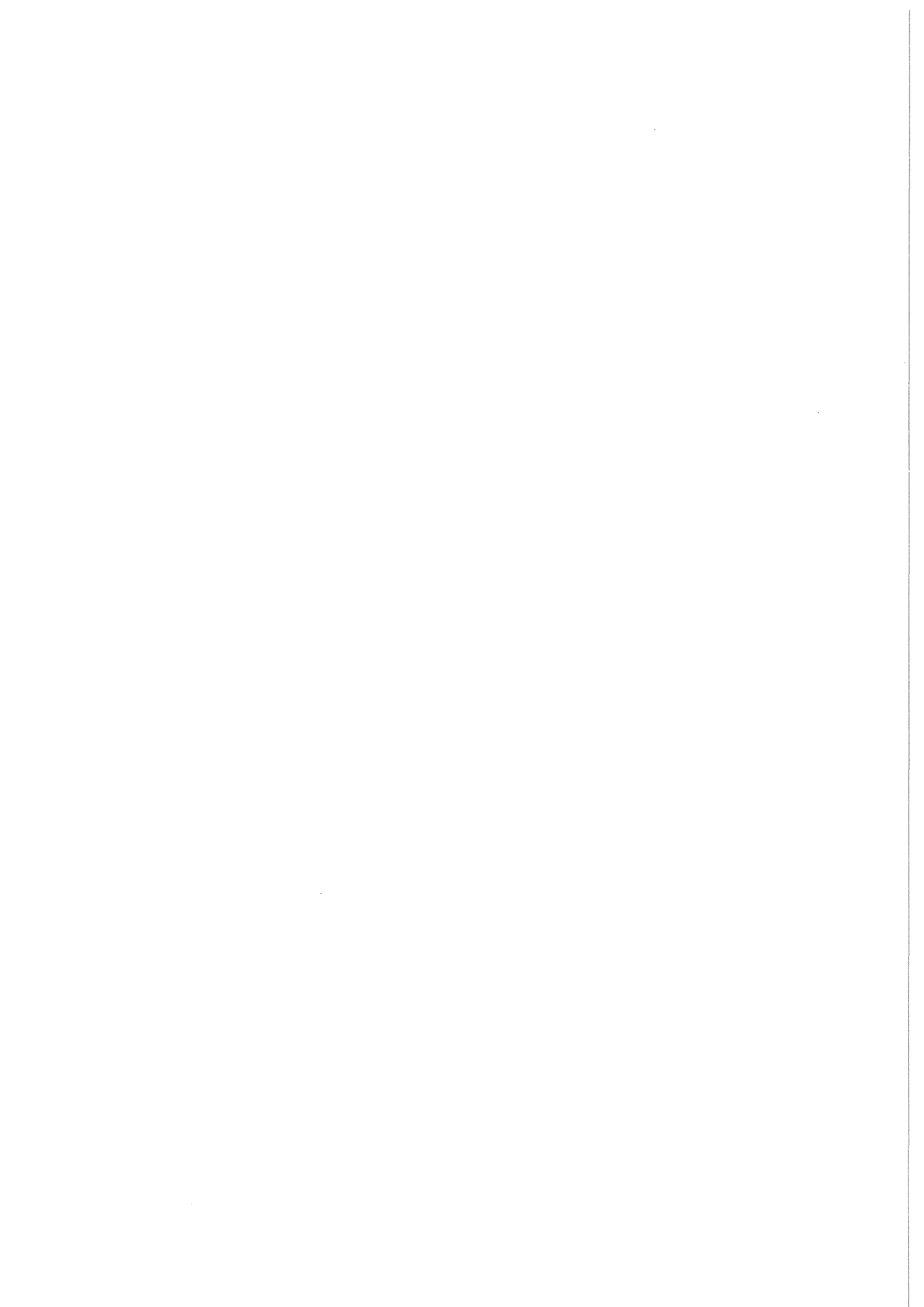


KfK 4370
Februar 1988

AHF
Ein menüorientiertes
Dialogsystem zur
Realisierung einer
anwendungsunabhängigen
Benutzerschnittstelle zu ISPF
Anwendungen unter MVS/TSO

W. Abel
Hauptabteilung Datenverarbeitung und Instrumentierung

Kernforschungszentrum Karlsruhe



KERNFORSCHUNGSZENTRUM KARLSRUHE

Hauptabteilung Datenverarbeitung und Instrumentierung

KfK 4370

AHF

Ein menüorientiertes Dialogsystem zur Realisierung
einer anwendungsunabhängigen Benutzerschnittstelle
zu ISPF Anwendungen unter MVS/TSO

W. Abel

Kernforschungszentrum Karlsruhe GmbH , Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

ZUSAMMENFASSUNG

In diesem Report wird eine Benutzerschnittstelle beschrieben, die es dem Anwender ermöglicht, ISPF-Anwendungen in einfacher Weise formal zu beschreiben und auszuführen. Die Auswahl der Anwendungen erfolgt über ein generiertes Master Menu. Die notwendigen Zuordnungen der Anwendungsdateien werden automatisch durchgeführt. Zur Verwaltung der Anwendungen wird die Tabellenverarbeitung des ISPF Dialog Management eingesetzt.

ABSTRACT

A menu-oriented dialog system to realise an application-independent user interface to ISPF applications under MVS/TSO

An interactive tool management interface, aimed at supporting the end-user in easily specifying and executing ISPF supported applications, is presented. As a result, an user related primary master menu may be generated as initialised by the interactive user. All necessary data set allocations and/or library concatenations with respect to the basic ISPF requirements have been done automatically after selection of an application from the master panel. This facility uses the ISPF Dialog Management table services for the administration of the application definition data.

INHALTSVERZEICHNIS

Einleitung	1
Anforderungen an eine ISPF-Benutzerschnittstelle	2
Formale Definition einer ISPF Anwendung	3
Organisation von ISPF-Anwendungen	3
Aufruf von ISPF-Dialoganwendungen	5
Realisierung einer Benutzerschnittstelle	6
<i>Lösungsansatz</i>	6
<i>Entwurf eines Benutzer-Interfaces zu ISPF-Anwendungen</i>	7
<i>Bemerkungen zur Aufrufunabhängigkeit der Schnittstelle</i>	8
<i>Vorteile</i>	9
<i>Nachteile, Beschränkungen</i>	10
Betrachtungen zur Performance des Interface-Programms	10
Aufruf der Benutzerschnittstelle	11
Die Dienste des Benutzer-Interface	12
Aufruf einer Anwendung	13
<i>Zuordnung der Anwendungsdateien</i>	13
<i>Aufrufbedingungen</i>	13
Initialisierung	13
Zuordnung von TSO/CLIST Dateien	14
Bearbeitung einer Anwendungsdefinition	15
<i>Hinzufügen einer Anwendung</i>	16
<i>Anpassung einer Anwendungsdefinition</i>	18
<i>Löschen einer Anwendung in der Tabelle</i>	19
Display-Funktionen	20
Eintragen von zusätzlichen Benutzerdateien	21
Fehlerbehandlung	22
<i>Eingabefehler</i>	22
<i>Dialogfehler</i>	22
<i>Systemfehler</i>	22
Tabellen der Benutzerschnittstelle	24
<i>Allgemeine Bemerkungen zur Tabellenverarbeitung</i>	24
<i>Aufbau der Tabellen der Benutzerschnittstelle</i>	24
<i>Installation der öffentlichen Anwendungs-Tabellen</i>	27
<i>Speicherung und Sicherung von Anwendungsdefinitionen</i>	27
Zusammenfassung und Ausblick	28
Anhang A: Aufruf von ISPF-Anwendungen ohne Benutzer-Interface	30
Anhang B: Verteilung der Zuordnungen mit Benutzer-Interface	30
Anhang C: Start-CLIST für das Benutzer-Interface	31
Anhang D: ISPF-Dialogbausteine	32
Anhang E: Dynamische Zuordnung von ISPF-Dateien	33

ABBILDUNGSVERZEICHNIS

Abbildung 1.	Vom SPF zum ISPF	1
Abbildung 2.	Namenskonventionen	4
Abbildung 3.	ISPF Benutzer-Interface Überblick	5
Abbildung 4.	Arten von ISPF-Anwendungen.	6
Abbildung 5.	ISPF Benutzer-Interface: Prinzipieller Datenfluß	8
Abbildung 6.	Haupt-Panel der ISPF Benutzerschnittstelle	11
Abbildung 7.	Generiertes Master Menu	12
Abbildung 8.	SYSPROC-Verkettung	14
Abbildung 9.	Display der aktuellen Anwendungstabelle	16
Abbildung 10.	Neudefinition einer Anwendung	17
Abbildung 11.	Globale Änderung der Programmumgebung	18
Abbildung 12.	Modifikation einer Anwendung	19
Abbildung 13.	Liste der aktuellen Dateitabelle	20
Abbildung 14.	Definition von Benutzerdateien	21
Abbildung 15.	In-Process-Panel nach Beendigung der Recovery	23
Abbildung 16.	Öffentlicher Pool für ISPF-Anwendungen	26
Abbildung 17.	Speicherung der Anwendungen	27
Abbildung 18.	AHF-Pool-Verwaltung	28
Abbildung 19.	Beziehungen der DD-Namen zu den ISPF-Basiskomponenten	32
Abbildung 20.	Suchreihenfolge für ISPF-Dateien bei LIBDEF-Service	33

EINLEITUNG

In den letzten 10 Jahren hat die Zahl der on-line ausgeführten interaktiven *full-screen* Anwendungen dramatisch zugenommen. Auch wurden mehr und mehr Programmierwerkzeuge bereitgestellt, um solche Anwendungen bei ihrer Entwicklung, Kodierung, Dokumentation und ihrem Test zu unterstützen. So hat sich innerhalb der "IBM-Welt" das SPF (*Structured Programming Facility*) als allgemein einsetzbares Software-Tool insbesondere unter dem Gesichtspunkt der sogenannten *Strukturierten Programmierung* durchgesetzt. Seit der Einführung des SPF in seiner ursprünglichen Form hat sich dieser *full-screen* Service zu einem Werkzeug weiterentwickelt, das es dem Benutzer ermöglicht, eigene interaktive Anwendungen zu programmieren. So gibt es bereits viele Anwendungen, die als *full-screen* Dialoge die Möglichkeiten des seit 1982 eingeführten ISPF (*Interactive System Productivity Facility*) nutzen.

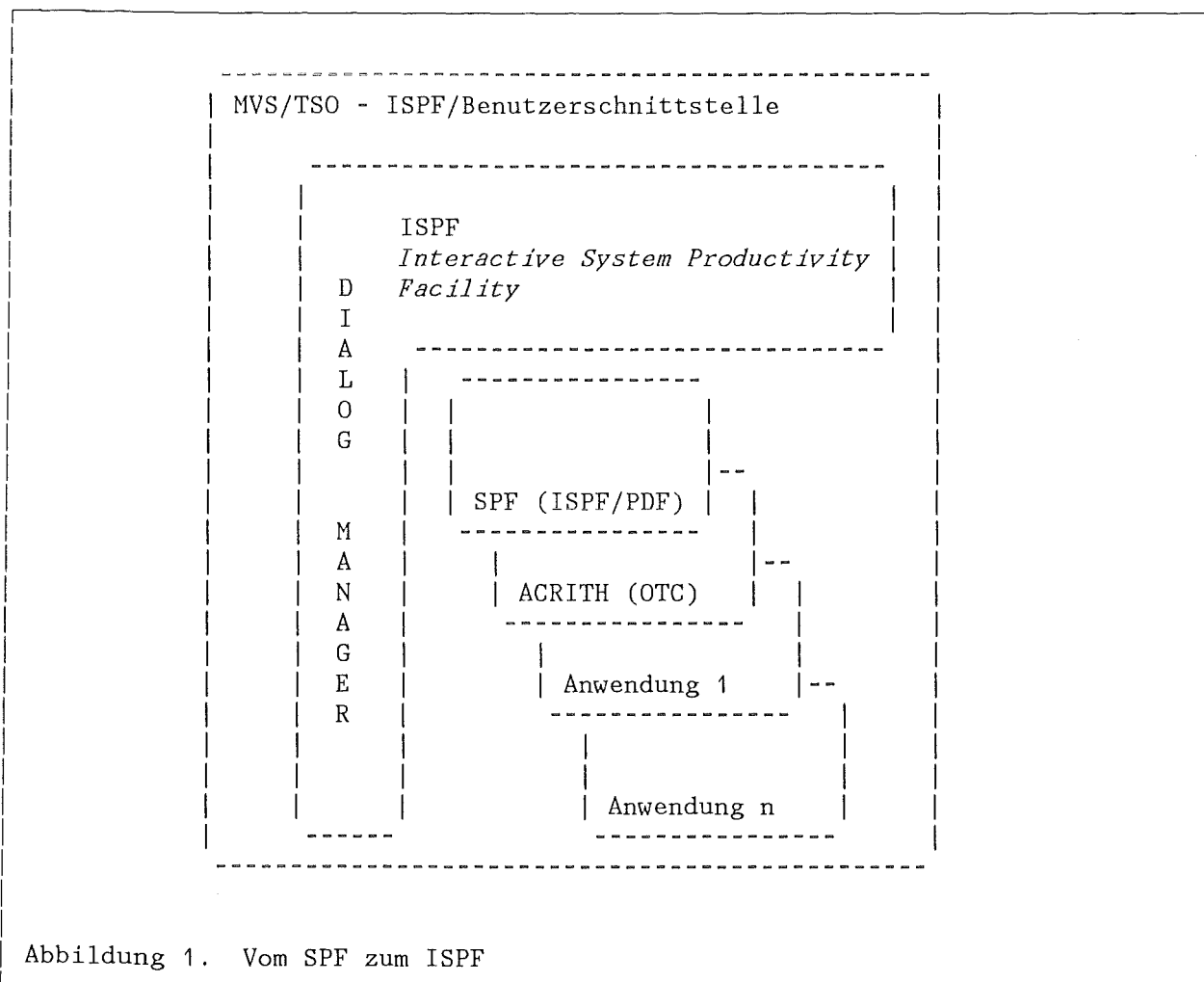


Abbildung 1. Vom SPF zum ISPF

Mit dem weiteren Ausbau wurden nicht nur neue Funktionen eingeführt, sondern es wird auch eine Zerlegung bestehender Funktionen in separate Dialogbausteine oder eigenständige Funktionen vorgenommen. Mit Hilfe dieser *Dialog Management Services* ist der Benutzer nun in der Lage, noch komplexere Dialoganwendungen zu entwerfen und je nach Eignung der Problemstellungen in interaktive Anwendungen umzusetzen. Daneben wurden und werden auch weiterhin von IBM bestehende Programmprodukte auf

ISPF umgestellt, bzw. neue Anwendungen bereitgestellt, die als ISPF-Dialoge geschrieben sind.

Abbildung 1 auf Seite 1 macht die oben beschriebene Situation deutlich. Sie zeigt insbesondere, daß das ursprüngliche SPF jetzt als selbständige Dialoganwendung aufgefaßt werden kann.

Im folgenden wird die Benutzerschnittstelle **AHF** (*Application Handling Facility*) beschrieben, die sowohl dem End-Benutzer in geeigneter Weise solche interaktiven Anwendungen zugänglich macht, als auch dem Entwickler die Möglichkeit gibt, neue Anwendungen zu definieren und zu testen. Hierzu sind die formalen Beschreibungen, wie Aufruf und benötigte Funktionsdateien, in ISPF-Tabellen gespeichert. Die Benutzerschnittstelle stellt eine in CLIST geschriebene Dialogfunktion dar, die es dem Benutzer gestattet, Operationen auf diese Tabellen durchzuführen und Werkzeuge aus dem so gewonnenen "Werkzeugkasten" auszuwählen, in einem generierten *Master Menu* zu kombinieren und auszuführen.

In diesem Sinne ist die Benutzerschnittstelle auch als ein Instrument anzusehen, daß mit gewissen Einschränkungen wegen des Fehlens einer allen Tools gemeinsamen Datenstruktur (Datenbank) auch in der *Anwendungsentwicklung* produktiv eingesetzt werden kann. Da das ISPF eine Schnittstelle zum APL2 besitzt, kommt dieser Sprache in diesem Zusammenhang eine besondere Bedeutung zu, da APL2 auch dem End-Benutzer eine Vielzahl von interaktiven Funktionen in einfacher, effizienter und produktiver Weise zugänglich macht. Neben dem ISPF als Dialog Manager steht so dem Benutzer über das APL2 in Kombination mit dem *Application Prototype Environment (APE)* ein weiteres mächtiges Werkzeug insbesondere zur Entwicklung von graphischen Anwendungen zur Verfügung.

Die Benutzeroberfläche soll der Realisierung einer *einheitlichen* Repräsentation aller ISPF-Anwendungen dienen. Neben solchen interaktiven Anwendungen können aber auch andere full-screen Dialoganwendungen verwaltet und aufgerufen werden, die nicht die ISPF Dialog Management Services verwenden.

Nach einer einführenden Charakterisierung einer ISPF-Benutzerschnittstelle wird die hier vorgeschlagene Realisierung in den Grundzügen beschrieben, um die gestellten Anforderungen zu erfüllen. Danach wird die Wirkungsweise des Dialogs anhand der dem Benutzer zur Verfügung gestellten Dienste im einzelnen erläutert.

Systemseitig ist die Implementierung des *ISPF V2R2M0* unter *TSO/E R2.1* (oder höher) unabdingbare Voraussetzung.

ANFORDERUNGEN AN EINE ISPF-BENUTZERSCHNITTSTELLE

Es ist allgemein bekannt, daß ein Dialog die Bereitstellung von Bildschirmmasken (Panels), Nachrichten, Dialogfunktionen, usw. erforderlich macht, um funktionsfähig zu sein. Diese *Dialogelemente* werden in Dateien geführt und müssen vor dem ISPF-Start unter bestimmten DD-Namen allokiert werden. Dies bedeutet aber, daß bei Änderung der Anforderungen jedesmal neue Zuordnungen getroffen werden müssen, sei es, daß Dateinamen sich ändern oder neue Anwendungen ausgeführt werden sollen. Anhang A zeigt in etwa eine mögliche Situation, wie ISPF-Anwendungen ausgeführt werden müssen.

Da außerdem Bildschirmmasken meist *statisch* definiert werden, muß auch hier eine Anpassung vorgenommen werden. Um danach einem Benutzer eine beliebige Anwendung

zugänglich zu machen und zwar direkt zur Ausführung, ohne ihn durch unnötiges Datenmanagement zu belasten, können in etwa die folgenden *Anforderungen an eine Benutzerschnittstelle* gestellt werden, die die Verwaltung von solchen interaktiven Anwendungen übernehmen sollte:

- Eine Bereitstellung von allgemein verwendbaren ("öffentlichen") ISPF-Anwendungen sollte möglich sein, ohne daß der Benutzer sich über die Organisation und Aufruf solcher Funktionen informieren muß.
- Die Zuordnung aller Anwendungen muß eindeutig sein, so daß sich die Anwendungen nicht gegenseitig behindern. Dies kann z.B. über die (möglichst strenge) Einhaltung von Namenskonventionen realisiert werden.
- Der Benutzer soll über die Benutzerschnittstelle in der Lage sein, sich eine interaktive Anwendung bei Bedarf zuzuordnen zu können, d.h. ohne "Programmierung" eine solche Anwendung zu aktivieren oder auch von der Verwendung wieder auszuschließen.
- Wartbarkeit und Benutzerfreundlichkeit
- Leichte Erlernbarkeit durch selbsterklärenden Dialog und Hilfsfunktionen
- Einfache Definition von Anwendungsumgebungen
- Unabhängigkeit von der Aufrufumgebung
- Autorisierung von Anwendungen
- Automatische Anpassung der Programmumgebung

FORMALE DEFINITION EINER ISPF ANWENDUNG

Eine ISPF-Anwendung läßt sich durch die folgenden Merkmale beschreiben:

1. *Auswahloption*
2. *Funktionsname (Kurztext)*
3. *Aufruf der Anwendung*
4. *Beschreibung (Langtext)*
5. *Applikationsparameter (NEWAPPL-Parameter)*
6. *Zugeordnete Dateien (Panels, Messages, ...)*
7. *Programmumgebung*

Die Aufgabe der zu entwickelnden Benutzerschnittstelle muß es demnach sein, diese Parameter in geeigneter Weise zu verwalten, wobei ein Benutzer, den obigen Forderungen entsprechend und durch die Dialogführung unterstützt, einen oder mehrere dieser Parameter leicht ändern kann.

ORGANISATION VON ISPF-ANWENDUNGEN

Allgemein verwendbare ISPF-Anwendungen sollten vor allem frei zugänglich zur Verfügung gestellt werden können, insbesondere auch ohne Notwendigkeit für die

Systemprogrammierung, solche Anwendungen bei Release-Änderungen o.ä. dem Standard-ISPF neu zuordnen zu müssen.

Zur Realisierung dieser Forderung können etwa folgende Maßnahmen getroffen werden.

- Einrichtung *einheitlicher* öffentlicher Bibliotheken für alle allgemein verwendbaren ISPF-Dialoge.

Diese Bibliotheken können von allen Benutzern gelesen, aber nur von der System/Anwendungsgruppe verändert werden. Eine Ausnahme kann hierbei in Sonderfällen die Tabellendatei sein.

- Um *gegenseitige Behinderungen* zu vermeiden, müssen alle Komponenten in den öffentlichen Bibliotheken strikten Namenskonventionen genügen. In Hinblick auf die Namensgebung des Standard-ISPF und anderen IBM-ISPF-Anwendungen, wie DCF oder RACF, kann das in Abbildung 2 angegebene Namensschema für die Vergabe von *Membernamen* benutzt werden.

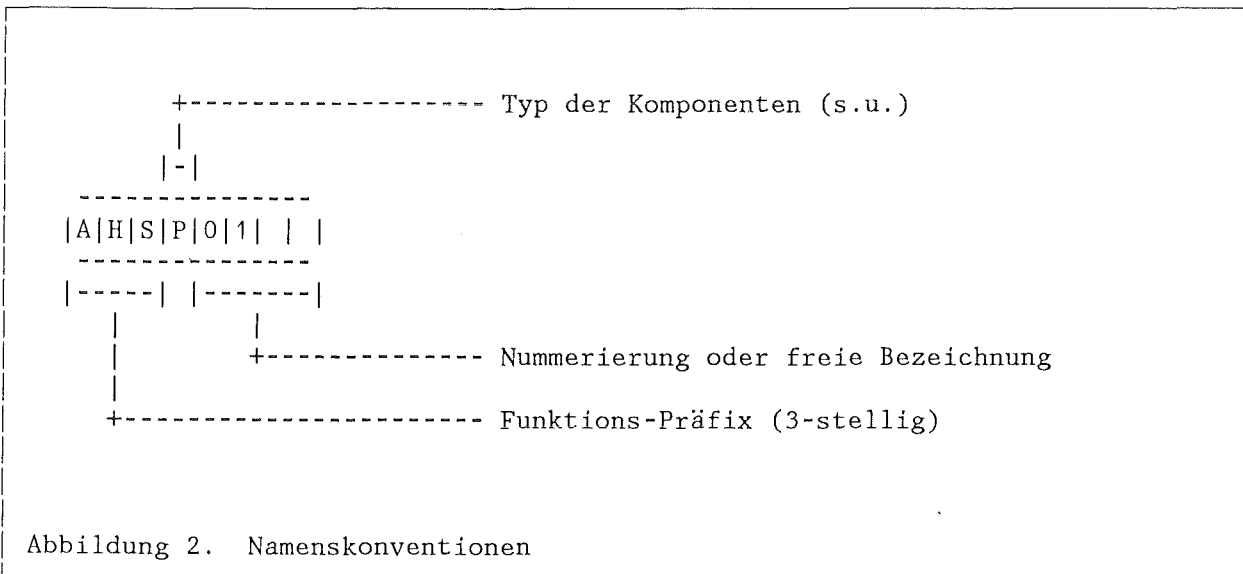


Abbildung 2. Namenskonventionen

Hierin sollte auf jeden Fall der Funktions-Präfix fest vorgegeben werden; die übrigen Positionen könnten frei gewählt werden. Zur besseren Unterscheidung kann jedoch der *Typ der Komponenten* verwendet werden:

- P Panels
- S Modelldatei
- M Messages
- T Tabellen
- C CLIST
- L Programme (Loadmodule)
- H Tutorial-Panel
- V Variable im Shared Pool oder Profile Pool

Die Namen der Messages können wegen der besonderen Definition nur maximal 7 Zeichen lang sein.

Die Angabe eines Funktions-Präfixes erfordert allerdings eine zentrale Registrierung bzw. Absprache. Die Standard-Präfixe des ISPF/PDF sind *ISP* bzw. *ISR*.

- Zuordnung der Dateien über die LOGON-Prozeduren oder Start-Prozeduren. Die Aufgabe der Benutzerschnittstelle liegt aber gerade in der Verwaltung dieser Dateien einer bestimmten Anwendung.

AUFRUF VON ISPF-DIALOGANWENDUNGEN

Wie aus den obigen Bemerkungen hervorgeht, müssen also alle Anwendungsdateien *vor dem Starten des ISPF* allokiert sein. Wurden die ISPF-Basis-Dateien bereits im LOGON zugeordnet, so ist diese Zuordnung wieder aufzulösen und eine Verkettung mit den Anwendungsdateien durchzuführen (siehe Anhang A).

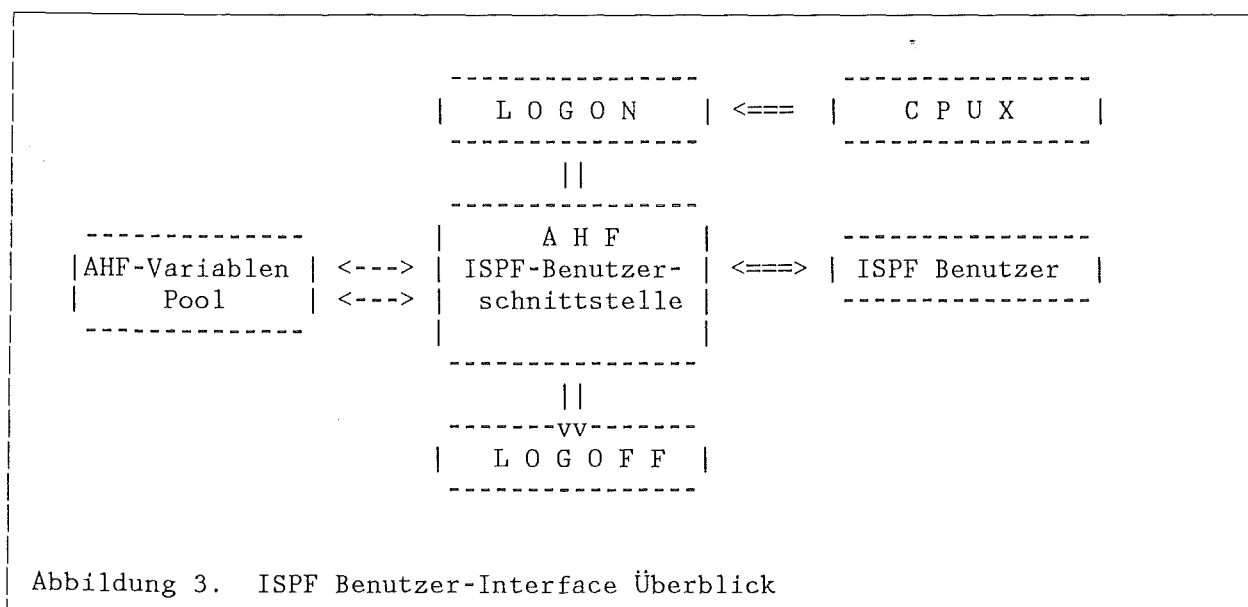


Abbildung 3. ISPF Benutzer-Interface Überblick

Andererseits ist aber auch eine Konstellation möglich, bei der im LOGON über die JCL oder die LOGON-CLIST *CPUX* keine Allokation von ISPF-Dateien ausgeführt wird. Dies würde zu einer *Vielzahl von Start-Prozeduren* führen, die nur umständlich zu verwalten wären und außerdem nur die Wartezeit bis zum Start der Anwendung ungünstig verschieben würden. Ebenso könnte die LOGON-CLIST alle Allokationen übernehmen. Für welche Lösung man sich entscheidet, hängt nicht zuletzt von Performance-Überlegungen oder von der Vielfältigkeit der zur Verfügung stehenden ISPF-Anwendungen ab bzw. wird auch von der Organisation des Betriebsablaufes einer Installation bestimmt.

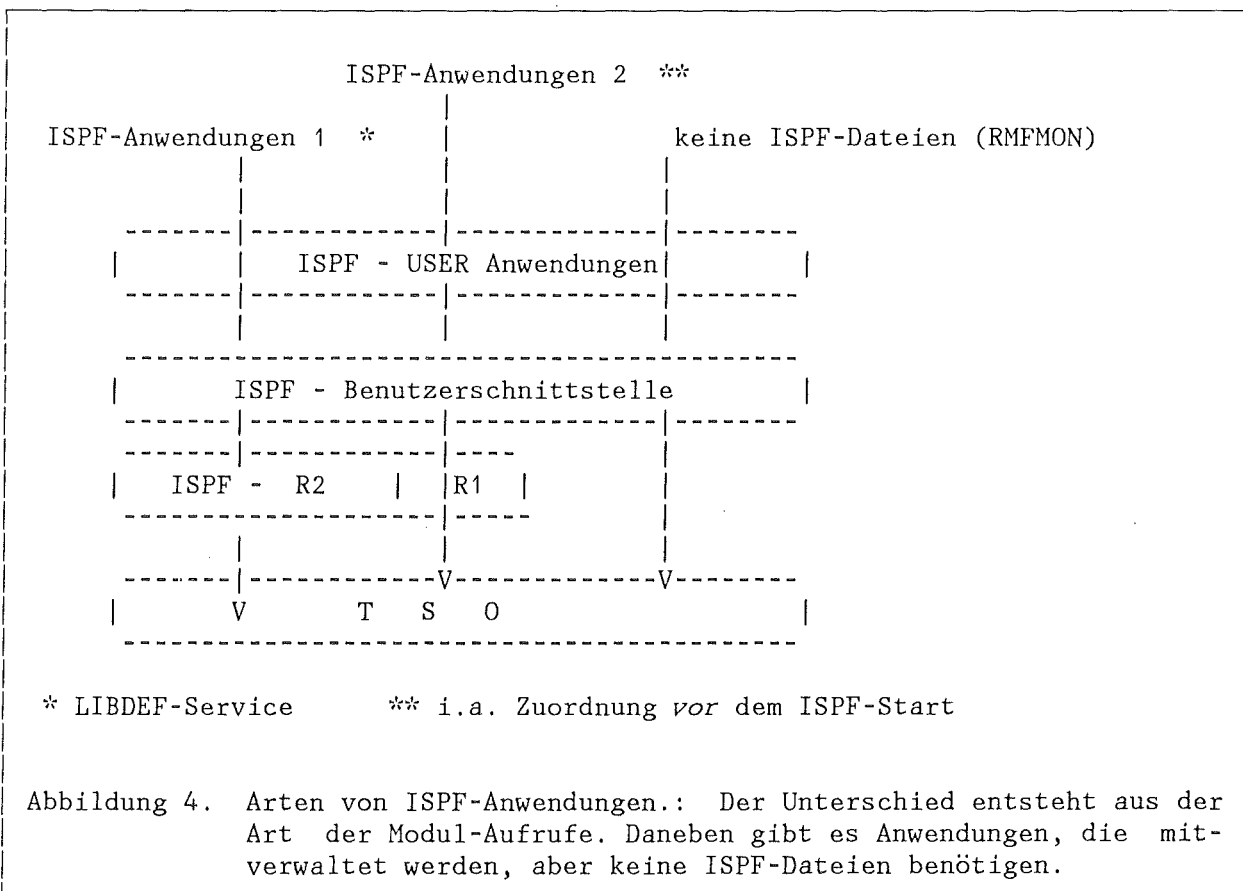
In unserem Fall wird die Zuordnung der ISPF-Basis-Dateien im LOGON durchgeführt; die Aktivierung der Anwendungen wird vollständig von der Benutzerschnittstelle übernommen (siehe Abbildung 3). Da diese Zuordnung *dynamisch* getroffen wird, bleibt die Pre-Allokation der ISPF-Basis-Dateien davon vollkommen unberührt.

REALISIERUNG EINER BENUTZERSCHNITTSTELLE

Lösungsansatz

Die Benutzerschnittstelle für ISPF-Anwendungen kann innerhalb der sich dem Benutzer eigentlich als Anwendungsprogrammmschnittstelle darstellende Benutzeroberfläche im engeren Sinne durch die folgenden Elemente charakterisiert werden:

- Vollständige *Parametrisierung* des Master-Panels
- Beschreibung der Anwendungen in *ISPF-Tabellen*
- *Dynamische* Zuordnung der Anwendungsdateien
 - Panels, Msgs, usw. mit Hilfe des *LIBDEF-Service*
 - Verkettung von CLIST über *SVC 99* bzw. *ALLOCATE*
 - Umschalten der Programmumgebung
- Umschalten der Programmumgebung
- Anwahl der Anwendung über den *SELECT-Service*



Bei Anwahl einer Anwendung ist also gegebenenfalls eine dynamische Allokation der zugeordneten Bibliotheken auszuführen. Abbildung 4 zeigt, daß es dabei drei verschiedene Arten von ISPF-Anwendungen geben kann. Anwendungen vom Typ 2 können aber nur dann aufgerufen werden, wenn die zugehörigen Load-Dateien in die LINK-Liste übernommen wurden oder in der STEPLIB geführt werden.

Entwurf eines Benutzer-Interfaces zu ISPF-Anwendungen

Das Verwaltungsprogramm soll eine möglichst unabhängige, flexible Handhabung und Ausführung von ISPF-Anwendungen unter MVS/TSO erlauben, ohne daß z.B. ein End-Benutzer etwas über die Definition der Anwendung und deren Umgebung wissen muß. Ein Dialogsystem (dies gilt allgemein) sollte sich ferner durch eine leichte Wartbarkeit, Übersichtlichkeit und "ergonomische" Bildschirmmasken auszeichnen. Nicht zuletzt sind eine formale Kontrolle der Eingabedaten und, wenn möglich, deren Plausibilitätsprüfung für einen sicheren Ablauf der Anwendung von großer Bedeutung.

Die Benutzerschnittstelle ist als ISPF-Hauptdialog in *CLIST* geschrieben. Sie besteht im wesentlichen aus einer Tabellen-gestützten Dialogfunktion, die es dem Benutzer ermöglicht, ein Master-Panel zu erzeugen bzw. seinen Wünschen anzupassen. Auf die Programmierung soll an dieser Stelle nicht eingegangen werden. Im folgenden werden lediglich einige charakteristische Eigenschaften und Funktionen zusammengefaßt:

- Verwaltung mit Hilfe des *ISPF-Tabellenservice*.
- Dialogelemente (Member): 69 *CLIST*, 73 Panels, 34 Msgs, 4 Skeletons
- Speicherung der Definitionen in ISPF-Tabellen.
- Hinzufügen, Löschen, Listen, Kopieren und Ändern einer Anwendungsdefinition
- Anbindung von anwendungsbezogenen Benutzer-Dateien
- Automatische Umschaltung des *LIBDEF-Service* bei Benutzer-Dateien
- Dynamische Verknüpfung von *SYSPROC* mit Anwendungs-*CLIST*-Dateien
- Änderung des Programm-Environments
- Split-Fähigkeit der Benutzerschnittstelle. Hierbei sind allerdings gewisse Einschränkungen hinsichtlich globaler Eigenschaften, wie Initialisierung, Änderung der Programmumgebung, u.a, unumgänglich, bzw. erfordern wie bei der *SYSPROC*-Verkettung eine besondere Behandlung.
- Aufruf mit einer allgemeinen Start-*CLIST* außerhalb von ISPF (siehe Anhang C)
- Benutzung eines eigenen Variablen-Pools (*Aufruf mit NEWAPPL(AHS)*)
- Generierung eines Master Menu aus der aktuellen Anwendungstabelle
- Automatische Initialisierung des Master Menu

Die prinzipielle Funktionsweise des Benutzerschnittstellenprogramms ist in Abbildung 5 auf Seite 8 schematisch dargestellt. Aufruf und Benutzung werden in den folgenden Kapiteln erläutert. Mit dem HELP-Key kann ein (Übersichts-) Tutorial aufgerufen werden.

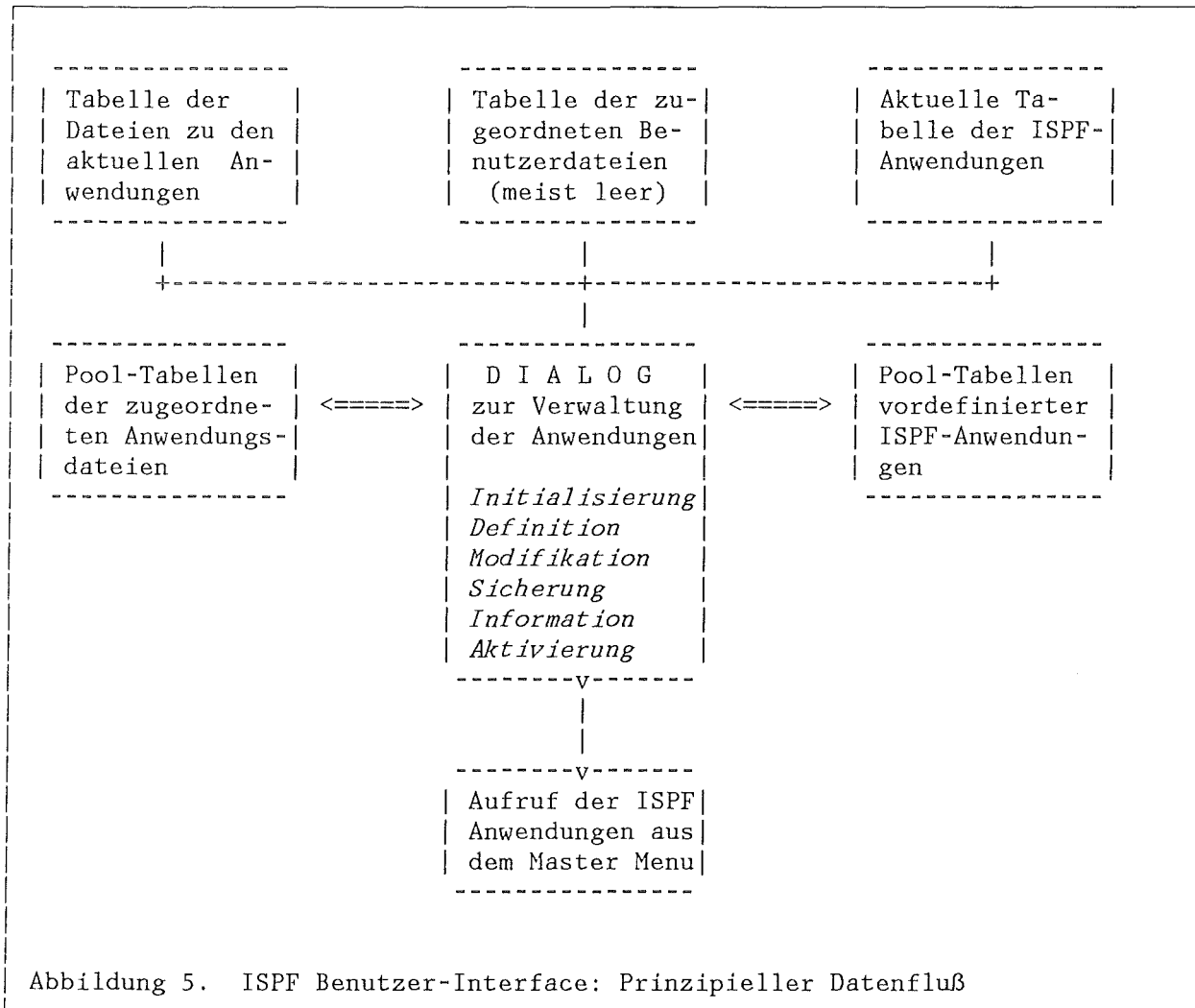


Abbildung 5. ISPF Benutzer-Interface: Prinzipieller Datenfluß

Bemerkungen zur Aufrufunabhängigkeit der Schnittstelle

Die Benutzerschnittstelle stellt einen eigenständigen ISPF-Dialog dar, der nur über ISPSTART aufgerufen werden kann. Eine der oben erhobenen Anforderungen an eine Benutzerschnittstelle bezieht sich darauf, daß der Benutzer nach dem Aufruf des AHF wieder seine alte Umgebung vorfindet, d.h. Zuordnungen von ISPF-Dateien sollten nicht verloren gehen. Hierzu werden vom AHF geeignete Maßnahmen getroffen, um diese Forderung möglichst einhalten zu können. Die Benutzerschnittstelle benutzt die Tabellenzuordnung des ISPF für den öffentlichen Anwendungspool und die Benutzertabellen. Diese Tabellen werden unabhängig in der separaten benutzerspezifischen Datei TABLES (siehe "Aufbau der Tabellen der Benutzerschnittstelle" auf Seite 24) geführt und müssen über die Files *ISPTLIB* bzw. *ISPTABL* vor dem ISPF-Start allokiert werden.

Anmerkung: Da diese Datei an erster Stelle in der Verkettung geführt wird, werden Tabellenausgaben in diese Datei geschrieben; Anwendungen sollten deshalb Ausgabetafeln nur mit Hilfe des LIBDEF-Service erstellen.

Weiterhin wird vom AHF die *SYSPROC*-Verkettung bearbeitet, so daß auch diese Zuordnung, wie sie vor dem Aufruf existierte, ausgelesen werden muß.

Beim Verlassen des AHF werden die ursprünglichen Pre-Allokationen der Files ISPTLIB, ISPTABL und SYSPROC wieder eingerichtet.

Ähnliches gilt auch für die *SYSLIB*-Verkettung, die u.U. von einer Anwendung oder vom Benutzer selbst über eine entsprechende Option geändert werden kann. Nach Beendigung einer Anwendung oder beim Verlassen des AHF wird die alte *SYSLIB*-Verkettung bzw. LOGON-Umgebung wiederhergestellt.

Zur (möglichst vollständigen) Wahrung der Unabhängigkeit der Benutzerschnittstelle von der Aufrufumgebung werden im folgenden noch einmal alle die Maßnahmen zusammengefaßt, die diese Forderung erfüllen helfen sollen:

- Dynamische Zuordnung der Anwendungsdateien
- Behandlung der ISPTLIB/ISPTABL-Verkettungen
- Behandlung der SYSPROC-Verkettung
- Wiedereinrichten der LOGON-Umgebung (*SYSLIB* und andere LOGON-Files)
- Aufruf des AHF mit eigenem PROFILE-Pool
- Führen der Tabellen in einer eigenen benutzerspezifischen Datei
- Unabhängige Installation, eigenständiges PP.

Vorteile

Aus den obigen Ausführungen ergeben sich vom Standpunkt des interaktiven Benutzers bei dem vorliegenden Konzept zur Realisierung einer Benutzerschnittstelle folgende grundlegende Eigenschaften:

- Bei LOGON oder vor dem ISPF Start muß keine Zuordnung von Anwendungsdateien ausgeführt werden. Dadurch kann ein schnelleres (LOGON und) Starten des ISPF erreicht werden.
- Durch eine automatische Anpassung der Programmumgebung ist die Aktivierung einer Anwendung von der LOGON-Prozedur unabhängig.
- Eindeutige Zuordnung von System- und Anwendungsdateien
 - Systemdateien bei LOGON bzw. beim ISPF Start (siehe Anhang B)
 - Anwendungsdateien bei Anwahl einer Anwendung
 - Geringerer Wartungsaufwand
 - Testumgebung für eine Anwendung ist (zusätzlich) leicht definierbar.

Nachteile, Beschränkungen

Aufgrund verschiedener Eigenschaften des Dialog Manager, die z.T. im Design dieses Tools zu suchen sind, müssen auch einige Einschränkungen in Hinblick auf die aktuelle Realisierung bisher in Kauf genommen werden:

- Bei Anwahl einer Anwendung können sich minimal schlechtere Antwortzeiten durch die dynamische Allokation der Anwendungsbibliotheken ergeben.
- Maximal 28 Anwendungen können in einem Master Menu geführt werden. Durch Austausch von Anwendungen zwischen der aktuellen Anwendungstabelle und dem Benutzerpool (siehe Abbildung 18 auf Seite 28) können jedoch so indirekt noch mehr Anwendungen verwaltet und aufrufbar gemacht werden.
- Pro Anwendung werden bisher nur 1 Datei pro ISPF-Dateityp verwaltet. Hier ist eine Erweiterung nur über ein zusätzliches Panel oder eine tabellengesteuerte Eingabe möglich.
- Bisher ist keine dynamische Definition von Nicht-ISPF-Dateien einer Anwendung vorgesehen. Eine solche Erweiterung wäre durch eine weitere Auswahloption im Definitionspanel einer Anwendung erreichbar (siehe auch "Aufbau der Tabellen der Benutzerschnittstelle" auf Seite 24).
- Im Split-Screen-Modus kann die Anwendungstabelle i.a. nicht geändert werden, weil hierzu ein *refresh* des Master Menu bei SWAP notwendig wäre, um wieder die richtige Zuordnung zu bekommen.
- Die Behandlung von ABEND's ist trotz des CONTROL Service bei Subtask-Fehlern nur mit größerem Aufwand zu lösen (siehe "Systemfehler" auf Seite 22). Der Dialog sollte in der Lage sein, nach einem ABEND geöffnete Dateien zu schließen, um ein erneutes LOGON zu vermeiden.

BETRACHTUNGEN ZUR PERFORMANCE DES INTERFACE-PROGRAMMS

Das Benutzerschnittstellenprogramm ist vollständig als CLIST geschrieben. Abgesehen von dem bereits erwähnten LIBDEF-Service des Dialog Management wird die Performance des Dialogs auch von umfangreichen Tests beeinflusst, wie Abprüfen auf Dateien, Prüfen auf Syntax der Anwendungsdefinitionen, und vor allem durch die Zuordnung von Anwendungsdateien mit dem ALLOCATE-Kommando. Trotz umfangreicher Maßnahmen, die zur Ablaufsicherheit des Dialogs beitragen können, sind die Antwortzeiten unter MVS/XA TSO/E Release 2.1 am Terminal vom Standpunkt eines interaktiven Benutzers aus gesehen ausreichend gut. In Hinblick auf die Wartbarkeit konnte so auf die Verwendung der Assembler-Sprache oder einer anderen höheren Programmiersprache, wie PL/1, verzichtet werden.

Da i.a. eine CLIST, wenn nicht anders vereinbart, als eine Transaktion aufgefaßt wird, wird eine solche CLIST insbesondere bei längerer Laufzeit (vor allem durch TSO-Kommandos, wie oben erwähnt) bei entsprechender Definition der Lastverteilung innerhalb der TSO *Performance Group* das Antwortzeitverhalten spürbar beeinflussen. Aus diesem Grund wurde ein TSO-Kommando entwickelt, das über *SYSEVENT*-Aufrufe an sinnvoller Stelle die Performance Group zurücksetzt, so daß dann wieder in der ersten Periode begonnen wird.

AUFRUF DER BENUTZERSCHNITTSTELLE

Das Benutzerschnittstellenprogramm wird mit der in Anhang C angegebenen Start-CLIST aufgerufen.¹ In der Start-Dialogfunktion wird geprüft, ob die Tabellendatei TABLES bereits Benutzer-Anwendungstabellen enthält, oder nicht. Ist die Datei noch leer, so wird die unten beschriebene *Initialisierung* durchgeführt. Hierzu wird die entsprechende Option in das Hauptpanel eingetragen. Die Terminaltastatur bleibt dann bis zur Aktivierung dieser Maßnahme blockiert. Im anderen Fall wird das *Master Menu* direkt aktiviert (siehe Abbildung 7 auf Seite 12). Es erfolgt also keine zwischenzeitliche Ausgabe des Haupt-Panels.

Soll die Zusammenstellung im Master Menu geändert werden, so muß der Benutzer durch Drücken der PF3-Taste in das Haupt-Panel zurückkehren, wo dann über die betreffenden Optionen die eigentlichen "*User-Application and Library Concatenation Services*" angewählt werden können.

```
***** ISPF APPLICATION HANDLING FACILITY *****
*****                                     *****
                                     KFK KARLSRUHE
                                     *****

SYSTEM - ISPF 2.2.0 MVS/XA TSO/E 2.1

User-Application and Library Concatenation Services

OPTION ==> _

USERID - ADI352
LOGON - FORTRAN

0 - Set AHF Options or ISPF Parameters
1 - Execute the Current ISPF/User Application Menu
2 - Define or Change Application Requirements
3 - Define Additional User Application-Level Libraries
4 - Initialise from Public Application Pool
5 - Define Additional CLIST Libraries for File SYSPROC Base
6 - Display/Print Utilities for Applications
7 - Invoke Application Pool Management Services
8 - Change User Program Environment (LOGON procedure)
I - News (new or changed public applications)
X - Terminate the ISPF Dialog using the List/Log Defaults

Press END Key (PF3) to terminate ISPF, press ENTER to start your Master Menu

Abbildung 6. Haupt-Panel der ISPF Benutzerschnittstelle
```

¹ In der KFK wird die Benutzerschnittstelle über das Kommando **SPF** aufgerufen.

DIE DIENSTE DES BENUTZER-INTERFACE

Das in Abbildung 6 dargestellte Panel zeigt die Möglichkeiten der Bearbeitung der Benutzeranwendungen auf, die die Forderung nach einer flexiblen Handhabung erfüllen sollen. Aus diesem Panel heraus können Anwendungen ausgeführt, definiert und initialisiert werden. Wahlweise kann bei Bedarf eine zusätzliche Zuordnung von anwendungsbezogenen Benutzerdateien vorgenommen werden.

- Aufruf des generierten Auswahlmenüs
- Bearbeitung der Anwendungsdefinitionen
- Definition von zusätzlichen anwendungsbezogenen Dateien
- Initialisierung aus dem öffentlichen Anwendungs-Pool
- Zuordnung von CLIST-Dateien
- Ausgabe der Tabelleninhalte
- Hilfsfunktionen zur Verwaltung der Pool-Tabellen
- Änderung der Programmumgebung (LOGON-Prozedur)

Da das ISPF/PDF automatisch der Anwendungsliste hinzugefügt wird, kann der Benutzer von hier aus durch zweimaliges Drücken der ENTER-Taste leicht in das PRIMARY OPTION MENU des ISPF/PDF gelangen, ohne die zugeordneten Optionen explizit auswählen zu müssen.

```
----- USER APPLICATION SELECTION MENU -----
OPTION ==> _

      1  ISPF/PDF      - ISPF/Program Development Facility (SPF)
      3  SLR           - Service Level Reporter
      A  APL2          - A P L  2
      D  GML           - Formatting with GML/DCF
      G  GDDM          - Interactive Graphic Application Services
      L  LISTA         - Display TSO Allocation List (LISTA ST)
      M  ACRITH        - Online Training Component (OTC)
      Q  MVS           - Information/Systems /Management /MVS
      S  SMP/E         - SMP/E Dialogs
      X  EXIT          - Terminate ISPF Applications

                                           Current PROC ==> FORTRAN
Press ENTER Key      to start the ISPF/PDF (SPF)
Press END Key (PF3) to return to the APPLICATION SERVICES Panel      AHF
```

Abbildung 7. Generiertes Master Menu: Die Optionen 1 und X werden automatisch hinzugefügt.

AUFRUF EINER ANWENDUNG

Eine Anwendung wird über das *generierte Master Menu* ausgewählt. Da das Master-Panel vollständig parametrisiert ist und die Variablen im AHF-Profile-Pool gehalten werden, kann das Master Menu unmittelbar nach dem Start des Benutzer-Interface aufgerufen werden (siehe Abbildung 7).

Zuordnung der Anwendungsdateien

Bei Anwahl einer Anwendung wird je nach Stand des Datei-Indikators (siehe "Aufbau der Tabellen der Benutzerschnittstelle" auf Seite 24) die Allokation der Anwendungsdateien durchgeführt. Nach Beendigung einer Anwendung werden die zugeordneten Dateien wieder freigegeben, bis auf diejenigen, die im Zusammenhang mit dem LIBDEF LIBRARY-Service allokiert wurden. Solche Allokationen bleiben also bestehen, so daß beim nächsten Aufruf der gleichen Anwendung eine schnellere Aktivierung der Anwendung möglich ist. Der CPU-Zeitgewinn liegt dabei, auf den LIBDEF-Service bezogen, bei etwa 1:6 gegenüber dem LIBDEF DATASET-Service, bei dem bei jedem Aufruf eine neue "Allokation" ausgeführt wird.

Der LIBDEF DATASET-Service wird jedoch nur dann benötigt, wenn zusätzliche Benutzerdateien vor dem LIBDEF-Service angesprochen werden sollen. Abbildung 20 auf Seite 33 zeigt eine ausführliche Darstellung der *Suchreihenfolge* für die Anwendungsdateien und ISPF Basis-Dateien.

Aufrufbedingungen

In diesem Zusammenhang soll darauf hingewiesen werden, daß jede Anwendung mit ihrem *eigenen Applikations-Profile* aufgerufen werden sollte, um gegenseitige Behinderungen auf Variablenebene zu vermeiden. Es sei denn, wie schon erwähnt, daß bei der Organisation von eigenen Anwendungen auf die strikte Einhaltung von Namenskonventionen geachtet wird. Dies ist i.a. aber nicht realistisch, so daß der Benutzer zu entscheiden hat, unter welchem Profile die oder mehrere Applikationen ausgeführt werden sollen. Wird bei Definition kein Applikations-Profile angegeben, so wird standardgemäß das Profile ISPPROF benutzt.

INITIALISIERUNG

Beim *ersten* Aufruf des Benutzer-Interface muß der Benutzer die entsprechenden Tabellen (die Tabellendatei selbst wird automatisch angelegt) aus einem "öffentlichen" Anwendungs-Pool in gewünschter Weise füllen, damit eine erste Definition eines Master-Panels ausgeführt werden kann. Dieser Vorgang wird beim ersten Aufruf *automatisch* ausgeführt. Die Zuordnung einer Anwendung zu einer Benutzergruppe wird dabei durch die bei der Definition festgelegte Authorisierung vorgenommen. In beiden Fällen wird jedem Benutzer ein vordefinierter Satz von Anwendungen zugeordnet (siehe Abbildung 16 auf Seite 26).

Bisher wird die Unterscheidung über die Zugehörigkeit des Benutzers zu einer bestimmten RACF-Gruppe vorgenommen. Hier könnte durch Setzen der entsprechenden Tabellenvariablen und formaler Einordnung eines Benutzers zu solchen so definierbaren Gruppen eine weitergehende Aufteilung der Anwendungen auf andere Benutzergruppen bzw. Fachbereiche durchgeführt werden. Danach wird das Master Menu unmittelbar ausgegeben. Bestehen die Tabellen bereits, so kann der Benutzer jederzeit die Initialisierung wiederholen.

Bei der ersten Initialisierung werden sowohl der Benutzer-Pool, als auch die Anwendungstabellen angelegt. Eine *wiederholte* Initialisierung geht dann i. a. vom Benutzer-Pool aus, wenn *kein Überschreiben des Benutzer-Pools* mit dem öffentlichen Pool zwecks Aktualisierung in einem hierzu ausgegebenen Panel gefordert wurde. In diesem Fall (und nur dann) gehen eigene Definitionen nicht verloren.

ZUORDNUNG VON TSO/CLIST DATEIEN

Wie schon oben erwähnt, kann der Benutzer im Bedarfsfall die SYSPROC-Verkettung ändern bzw. erweitern. Eine solche Umdefinition wird dann unabhängig von den Anwendungen als *neue Basis-Allokation* verwendet. Die in Abbildung 8 gezeigten Systemdateien können allerdings (bisher) nicht verändert werden. Es wird eine Prüfung des Rekordformats durchgeführt; VB-Dateien können nicht in der Verkettung benutzt werden.

```
----- EDIT/VERIFY TSO/CLIST LIBRARIES -----  
COMMAND ==>>>  
  
The following data sets have been predefined for SYSPROC :  
  
BASIC CLIST CONCATENATION ==> 'TSOSYS.ADICLIB'  
                           ==> 'TSOSYS.CMDPROC.CLIST'  
                           ==> 'TSOSYS.ISRCLIB'  
                           ==>  
                           ==>  
                           ==>  
                           ==>  
  
Enter additional data set names, where desired:  
  
GENERAL CLIST LIBRARY      ==> A.CLIST_  
  
EDIT MACRO CLIST LIBRARY  ==>  
  
Press ENTER to add, press END Key (PF3) to return without updating  
  
Abbildung 8. SYSPROC-Verkettung: Das Panel zeigt die Basis-Verkettung.  
Zusätzlich können weitere Benutzer-CLIST, hier A.CLIST, angegeben werden.
```

Zur dynamischen Anbindung von CLIST-Dateien steht *kein LIBDEF-Service* zur Verfügung. Um von etwaigen Pre-Allokationen vor dem Aufruf der Benutzerschnittstelle unabhängig zu sein, d.h. diese nicht zu überschreiben, wurde die Zuordnung von anwendungsbezogenen CLIST-Dateien über eine dynamische Verkettung mit Hilfe des *SVC 99* getroffen. Bei Beendigung einer Anwendung wurde diese Verkettung gelöst, so daß dadurch der ursprüngliche Zustand wiederhergestellt wurde. Da wegen der Eigenschaft einer solchen Verkettung die Benutzerdateien erst *nach* den Systemdateien (+ Pre-Allokationen) durchsucht werden, ergeben sich nicht nur aus Performance-Gründen, sondern auch aus möglichen Namensverwechslungen Probleme bei der Ausführung von öffentlichen Anwendungen, wenn diese zusätzlich als Testversion geführt werden sollen.

Da i.a. Benutzer-Dateien *vor* den Systemdateien durchsucht werden sollten, wurde die dynamische Verkettung durch das TSO/ALLOCATE ersetzt. Allerdings vergrößert sich dadurch der Verwaltungsaufwand nicht unerheblich. So muß die SYSPROC-Verkettung beim Aufruf der Benutzerschnittstelle analysiert und registriert werden. Hierzu kann das TSO/LISTA Kommando benutzt werden, dessen Ausgabe mit Hilfe der SYSOUTTRAP-Möglichkeit in einer CLIST untersucht wird.

Bei Anwahl einer Anwendung wird jeweils die gesamte Liste aller notwendigen CLIST-Dateien bereitgestellt und eine Allokation für SYSPROC ausgeführt; wird die Anwendung beendet, so muß die vorherige Allokation wiederhergestellt werden. Insbesondere gilt dies im *Split-Screen-Modus*, wo dann die CLIST-Datei der auf dem zweiten Schirm aktivierten Anwendung zusätzlich zu der bestehenden Verkettungsliste hinzugefügt werden muß; diese Allokation bleibt dann so lange bestehen, bis eine der Anwendungen beendet wird. Dieser Vorgang erfordert sicher notwendigerweise mehr Zeit als eine in ähnlicher Weise wechselweise auszuführende *concatenation* und *deconcatenation*.

Die sogenannte Basis-Verkettung kann vom Benutzer inspiziert und um zwei weitere allgemeine CLIST-Dateien, insbesondere für EDIT-MACROS, erweitert werden, indem über Option 5 im Haupt-Panel der Benutzerschnittstelle eine neue Basis-Verkettung für SYSPROC durchgeführt wird (siehe Abbildung 8 auf Seite 14).

BEARBEITUNG EINER ANWENDUNGSDEFINITION

Nach Auswahl der Option 2 des Service-Panels (siehe Abbildung 6 auf Seite 11) kann der Benutzer seine Anwendungstabelle bearbeiten. Auf dem Bildschirm erscheint dann die in Abbildung 9 auf Seite 16 gezeigte aktuelle Anwendungstabelle, in der die verschiedenen Bearbeitungsmöglichkeiten durch die dort vorgesehenen Zeilenkommandos ausgewählt werden können:

- Hinzufügen einer Anwendung
- Löschen einer Anwendung
- Modifikation einer Anwendungsdefinition
- Wiederholen einer Definition unter neuem Namen (Repeat)
- Listen der Dateien einer Anwendung
- Übertragen einer Definition aus dem Benutzer-Pool in die Tabelle
- Übertragen einer Definition aus dem öffentlichen Pool in die Tabelle
- Sichern einer Definition im Benutzer-Pool
- Ausführung einer Anwendung (Test)

Diese Tabelle kann mit Hilfe der Kommandos *SORTA* und *SORTD* bzgl. der Auswahloption aufsteigend bzw. absteigend sortiert werden. Die Kommandos *SORTFA* und *SORTFD* wirken auf die Funktionsnamen in der entsprechenden Weise.

```
----- ISPF/USER-RELATED APPLICATIONS ----- ROW 1 OF 8
COMMAND ==> _                                SCROLL ==> PAGE

The following line commands A are active:

A - Add new application after line      I - Insert from pool after line
D - Delete application                  C - Copy application to private pool
M - Modify application                  R - Repeat under new name
L - List application-related libraries  X - Execute application

A      ID      FUNCTION      DESCRIPTION
-----
|  A  |  APL2      |  A P L 2
|  D  |  GML       |  Formatting with GML/DCF
|  G  |  GDDM      |  Interactive Graphic Application Services
|  L  |  LISTA     |  Display TSO Allocation List (LISTA ST)
|  M  |  ACRITH    |  Online Training Component (OTC)
|  Q  |  MVS       |  Information/Systems /Management /MVS
|  S  |  SMP/E     |  SMP/E Dialogs
|  3  |  SLR       |  Service Level Reporter
***** BOTTOM OF DATA *****

Abbildung 9.  Display der aktuellen Anwendungstabelle:  Je nach  Aktionskode
              kann eine der angegebenen Maßnahmen getroffen werden.
```

Hinzufügen einer Anwendung

Zur Neudefinition einer Anwendung muß der Benutzer das in Abbildung 10 auf Seite 17 gezeigte Panel ausfüllen. Nach Drücken der ENTER-Taste erscheint wieder die Anwendungstabelle mit der nun eingetragenen Anwendung.

Anmerkung: Bisher kann jeweils nur eine Datei pro Dialogelement angegeben werden.

Wurden die Dialogelemente, wie Panels usw., bereits in "öffentlichen" Bibliotheken abgelegt und bei LOGON zugeordnet, so brauchen keine Dateinamen eingetragen zu werden.

Die Dateinamen können der TSO-Konvention entsprechend eingegeben werden. Nach Beendigung der Eingabe werden die Dateien auf ihre Existenz hin überprüft. Wurde eine Datei nicht gefunden, so wird ein *Fehler-Panel* ausgegeben, so daß der Benutzer die betreffenden Dateinamen ändern kann oder aber einfach, wenn sinnvoll, austreichen kann.

Eine Anwendung eintragen heißt i.a., daß eine solche Anwendung bereits existiert und auch ausführbar ist. Damit aber auch noch nicht entwickelte Anwendungen de-

finiert werden können, kann der Benutzer aus diesem Panel heraus der Reihe nach die vorher angegebenen Anwendungsdateien allokatieren und mit Hilfe der unter AHF zur Verfügung stehenden Softwareprodukte, wie ISPF/PDF, GDDM, Compiler, APL2 (APE) und GML, in diesem Sinne auch eine Anwendungsentwicklung betreiben.

Anmerkung: *Dieser Test wird allgemein ausgeführt.*

Ist die Korrektur eines Dateinamens nicht möglich, so kann durch Drücken der END(PF3)-Taste die Definition einer Anwendung verhindert werden.

```
----- NEW APPLICATION DATA ENTRY PANEL -----
COMMAND =====> _

SELECT OPTION           =====>
APPLICATION FUNCTION ID =====>
DESCRIPTION             =====>
DIALOG INVOCATION      =====>
APPLICATION PROFILE POOL =====>      ( blank to use the default profile)
PROGRAM ENVIRONMENT    =====>      ( blank if left asis)
AUTHORISATION CODE     =====>      ( blank or 0 for general use)

Enter data set names where necessary:

CLIST LIBRARY          =====>
PANEL LIBRARY          =====>
MESSAGE LIBRARY        =====>
SKELETON LIBRARY       =====>
TABLE INPUT LIBRARY    =====>
TABLE OUTPUT LIBRARY   =====>
FILE TAILORING OUTPUT  =====>
LOAD LIBRARY           =====>

Press ENTER to add the application, press PF3 to return without updating

Abbildung 10. Neudefinition einer Anwendung
```

Ist die Ausführung einer Anwendung von einer bestimmten **Programmumgebung** (PL1, FORTRAN, usw.) abhängig, so kann diese in Hinblick auf die LOGON-Prozedur *anwendungsspezifisch* zugeordnet werden. Bei Aufruf der Anwendung werden vor der eigentlichen Aktivierung die notwendigen Allokationen (SYSLIB-Verkettung und andere Files) vorgenommen. **Im allgemeinen sind aber Anwendungen so geschrieben, daß sie in einer beliebigen Programmumgebung ablaufen können.** Wenn gewünscht, kann die Programmumgebung auch global über die Option 8 des Haupt-Panels eingestellt werden (siehe Abbildung 11 auf Seite 18). Die zugeordneten Dateinamen sind in der öffentlichen Tabelle gespeichert und werden zentral verwaltet. Im folgenden sind die Parameter für die verschiedenen Definitionsmöglichkeiten in Anlehnung an die KFK-LOGON-Prozeduren aufgeführt:

- 0 beliebig (die LOGON-Prozedur wird nicht geändert)
- 1 GENERAL
- 2 FORTRAN (FORTRAN 77)
- 3 VSFORT (VS FORTRAN)
- 4 PL1

5 ASM

Bei Beendigung einer Anwendung wird die LOGON-Umgebung wiederhergestellt. Im *Split-Screen-Modus* ergibt sich dabei jedoch eine besondere Situation, weil dann die SYSLIB-Verkettung der auf dem zweiten Schirm zu aktivierenden Anwendung mit der bestehenden Verkettung verglichen werden muß; ist diese Programmumgebung von der auf dem ersten Schirm aktiven unterschiedlich, so muß die Ausführung dieser Anwendung unterdrückt werden.

```
----- PROGRAM ENVIRONMENT SELECTION PANEL -----
OPTION ==> _

CURRENT PROCEDURE ==> PL1          LOGON PROCEDURE ==> FORTRAN

Select one of the options below:

1 - GENERAL                5 - ASSEMBLER
2 - FORTRAN                6 - COBOL
3 - VSFORT                 7 - PASCAL
4 - PL1                    8 - VPFORT

D - Show Current User Program Environment
M - Modify the Program Environment Definition (for authorised users only)

The selected option will be used to establish a special program environment
with respect to files, such as SYSLIB or other files for running an appli-
cation. This option is related to the LOGON-procedure as chosen by the user.

Press PF3 key to return without changing the program environment

Abbildung 11. Globale Änderung der Programmumgebung
```

Wird dann z.B. die auf Schirm 1 aktive Anwendung, die bereits die vom LOGON her bestimmte Programmumgebung geändert hatte, beendet, so kann dann auf Schirm 2 die vorher verbotene Anwendung ausgeführt werden.

Allerdings muß dabei in Kauf genommen werden, daß bei SWAP auf dem anderen Schirm vorerst nicht die richtige Programmumgebung angezeigt wird, da beim Schirmwechseln kein *refresh* der Panel-Inhalte durchgeführt werden kann.

Anpassung einer Anwendungsdefinition

Im folgenden wird beschrieben, wie eine Anwendungsdefinition modifiziert werden kann. Dabei lassen sich alle Einträge verändern. Nach erfolgreichem Test und Überführung der Dateien in öffentliche ISPF-Basisdateien kann eine Anwendung durch Wegstreichen der Datei-Einträge als öffentliche Anwendung weiterhin benutzt

werden, oder mit Hilfe der REPEAT-Funktion *parallel* als Test-Version unter *neuem* Namen und neuer Auswahloption weitergeführt werden.

Anmerkung: Die Funktionsnamen müssen eindeutig sein, da sie als Schlüsselbegriff in den Tabellen benutzt werden!

Ist eine Anwendung bereits öffentlich, d.h. werden die Anwendungsdateien in einer ISPF-Basis-Datei geführt, so kann durch den Eintrag einer Datei eine Test-Version leicht hergestellt werden, wenn z.B. eine erweiterte Anbindung einer Anwendung über ein neues Eingangspanel getestet werden soll.

Beim Aufbau einer Test-Version, zusätzlich zu einer bestehenden öffentlichen Version, könnte der Applikationsparameter geändert werden, damit die PROFILE-Variablen nicht überschrieben werden und so beide Versionen auch auf Variablenebene möglichst ohne Seiteneffekte nebeneinander ausgeführt werden können.

```
----- MODIFICATION ENTRY PANEL -----
COMMAND =====> _

APPLICATION =====> SLR                APPLICATION PROFILE POOL =====> SLR
                                           PROGRAM ENVIRONMENT           =====> 0
SELECT OPTION          =====> 3        AUTHORISATION CODE             =====> 0
FUNCTION ID            =====> SLR
DESCRIPTION            =====> Service Level Reporter
DIALOG INVOCATION     =====> PANEL(DRE@PRIM)

Change data set names where desired:

CLIST LIBRARY          =====> 'TSOSYS.V2R2.SLRCLIST'
PANEL LIBRARY          =====> 'TSOSYS.V2R2.SLRPANEL'
MESSAGE LIBRARY        =====> 'TSOSYS.V2R2.SLRMSGS'
SKELETON LIBRARY       =====>
TABLE INPUT LIBRARY    =====>
TABLE OUTPUT LIBRARY   =====>
FILE TAILORING OUTPUT  =====>
LOAD LIBRARY           =====> 'TSOSYS.V2R2.SLRLOAD'

Press ENTER to modify, press PF3 to return without changing

Abbildung 12. Modifikation einer Anwendung
```

Löschen einer Anwendung in der Tabelle

Wird eine Anwendung in der Applikationstabelle gelöscht, so werden in den zugeordneten Library- und Benutzerdatei-Tabellen etwaige Einträge automatisch gelöscht. Dabei wird, ebenso wie beim Hinzufügen von Benutzerdateien oder deren Elimination, der Datei-Indikator (siehe auch "Aufbau der Tabellen der Benutzerschnittstelle" auf Seite 24) automatisch angepaßt, so daß beim Aufruf der Anwendung der richtige LIBDEF-Service ausgewählt werden kann.

```
----- ALL APPLICATION-LEVEL LIBRARIES ----- ROW 1 OF 18
COMMAND ==> _                               SCROLL ==> PAGE
```

APPL	DATA SET NAME	FILE	TYP
ACRITH	'TSOSYS.DCQMLIB'	ISPMLIB	M
ACRITH	'TSOSYS.DCQOLIB'	ISPMLIB	L
ACRITH	'TSOSYS.DCQCLIST'	SYSPROC	C
ACRITH	'TSOSYS.DCQPLIB'	ISPPLIB	P
MVS	'TSOSYS.INFO.V2R1M0.BLGMAC2'	ISPPLIB	P
MVS	'TSOSYS.INFO.V2R1M0.LOAD'	ISPMLIB	L
SLR	'TSOSYS.V2R2.SLRMSG'	ISPMLIB	M
SLR	'TSOSYS.V2R2.SLRCLIST'	SYSPROC	C
SLR	'TSOSYS.V2R2.SLRLOAD'	ISPMLIB	L
SLR	'TSOSYS.V2R2.SLRPANEL'	ISPPLIB	P
SMP/E	'TSOSYS.SMPTABL'	ISPTABL	O
SMP/E	'TSOSYS.SMPTABL'	SMPTABL	A
SMP/E	'TSOSYS.SMPTABL'	ISPTLIB	I
SMP/E	'SYS1.SGIMTBLO'	ISPTLIB	I
SMP/E	'SYS1.SGIMLMDO'	ISPMLIB	L
SMP/E	'SYS1.SGIMPNO'	ISPPLIB	P
SMP/E	'SYS1.SGIMPRCO'	ISPSLIB	S
SMP/E	'SYS1.SGIMMSGO'	ISPMLIB	M

***** BOTTOM OF DATA *****

Abbildung 13. Liste der aktuellen Dateitabelle

DISPLAY-FUNKTIONEN

Die Inhalte der Anwendungstabelle und der zugeordneten Datei-Tabellen können auf dem Bildschirm dargestellt oder über einen generierten SUBMIT-Job auf einem Drucker ausgegeben werden:

- Anzeige der aktuellen Anwendungstabelle
- Darstellung der zugeordneten Dateitabelle
- Anzeige der anwendungsbezogenen Benutzer-Dateien (meist leer)
- Darstellung des Benutzer-Pools
- Information über die öffentlichen Anwendungen
- Hardcopy der aktuellen Anwendungsdefinitionen
- Anzeige der aktuellen Programmumgebung

In Abbildung 13 wird als Beispiel eine aktuelle Liste der den ISPF-Anwendungen zugeordneten Anwendungsdateien gezeigt.

EINTRAGEN VON ZUSÄTZLICHEN BENUTZERDATEIEN

Sollen neben den Anwendungsdateien noch weitere Benutzerdateien zur Beschreibung einer ISPF-Anwendung herangezogen werden, so kann eine solche Zuordnung durch die Option 3 in Abbildung 6 auf Seite 11 eingeleitet werden. Diese Dateien werden vom Benutzerschnittstellenprogramm bei Anwahl der betreffenden Anwendung *vor* der Ausführung des LIBDEF-Service allokiert und nach Beendigung der Anwendung wieder freigegeben. *Die Dateinamen werden in der Tabelle USRTBL der TABLES-Datei geführt.*

Bevor die Daten eingetragen werden können (siehe Abbildung 14) wird vorher nach der betreffenden Anwendung gefragt und diese daraufhin untersucht, ob Funktionsdateien überhaupt definiert sind, oder nicht. Im letzteren Fall ist *keine* Definition von solchen anwendungsbezogenen Dateien möglich, da die Ausführung eines LIBDEF-Service hierzu Voraussetzung ist.

```
----- APPLICATION RELATED USER DATA SETS -----  
COMMAND =====> _  
  
APPLICATION =====> SLR  
  
Enter data set names where desired:  
  
USER PANEL LIBRARY           =====>  
USER MESSAGE LIBRARY         =====>  
USER SKELETON LIBRARY         =====>  
USER TABLE INPUT LIBRARY     =====>  
USER TABLE OUTPUT LIBRARY    =====>  
USER FILE TAYLORING OUTPUT    =====>  
USER LOAD LIBRARY             =====>  
  
Press ENTER to add, press PF3 to return without updating  
  
Abbildung 14. Definition von Benutzerdateien: Solche Dateien können tempo-  
rär zugeordnet werden und werden vor dem LIBDEF-Service  
durchsucht.
```

Wenn ein Eintrag für eine Anwendung gemacht wurde, so wird im Falle einer Initialisierung (Option 4) automatisch der Dateiindikator angepaßt, so daß solche Dateien beim Aufruf der betreffenden Anwendung wieder zugeordnet werden können.

FEHLERBEHANDLUNG

Insbesondere beim Test neuer Anwendungen können verschiedene Fehler auftreten, die sich je nach Schwierigkeitsgrad in die folgenden drei Kategorien einteilen lassen:

Eingabefehler

Eingabe- oder sonstige Verarbeitungsfehler sollten von der betreffenden Dialogfunktion ausreichend behandelt werden. Die Anwendung kann i.a. ohne Fehler entaktiviert und auch wieder aufgerufen werden.

Dialogfehler

Dialogfehler bzgl. der Dialogelemente werden vom Dialog Management erkannt und durch die Ausgabe eines Fehlerpanels signalisiert. Fehlt z.B. eine Paneldefinition, so wird über den CONTROL-Service die Anwendung beendet und die Kontrolle an das AHF zurückgegeben. Das AHF-Fehlerpanel zeigt die ISPF-Fehlermeldung; der Benutzer kann durch Drücken der ENTER-Taste ins Auswahlnenü (Master Menu) zurückkehren. Vorher werden die Allokationen der fehlerhaften Anwendung, wie im Normalfall, bis auf geöffnete Dateien aufgelöst.

Systemfehler

Der schwerwiegendste Fehler liegt vor, wenn die Dialogfunktion mit einem ABEND endet. Wird z.B. ein Modul nicht gefunden, so bekommt man einen ABEND 806. Dies kann daran liegen, daß die allokierte Load-Datei das gesuchte Member überhaupt nicht enthält, oder aber die Load-Datei als STEPLIB geführt werden muß. Im letzteren Fall kann eine solche Situation, wie schon oben erwähnt, dadurch vermieden werden, daß die betreffende Datei in die LINK-Liste übernommen wird.

Im Falle eines ABEND wird das ISPF mit einem Subtask-Fehler abgebrochen, ohne dabei die Kontrolle an den Dialog zurückzugeben. Mit Hilfe des CONTROL SUBTASK PROTECT-Service bekommt der Dialog dahingegen die Kontrolle indirekt zurück, indem das ISPF ein Fehlerpanel mit dem ABEND-Kode ausgibt. Nach Drücken der ENTER-Taste wird ein AHF-Fehlerpanel ausgegeben, das die abgebrochene Anwendung zeigt. Mit der ENTER-Funktion kann der Dialog fortgesetzt werden: Der Dialog wird neu aufgesetzt, d.h. er beginnt wieder bei der Start-Dialogfunktion. Damit aber nach Möglichkeit ein eindeutiger Zustand hinterlassen bzw. wiederhergestellt wird, muß RECOVERY ON über die Option 0 des Haupt-Panels gesetzt worden sein. Es werden im einzelnen folgende Maßnahmen getroffen:

- Bei einem ABEND beginnt der Dialog also wieder von vorn, jedoch haben die Profile-Variablen nicht mehr die aktuelle Belegung, da das AHF-Profile offensichtlich neu eingelesen wird (nicht dokumentiert). Damit der aktuelle Zustand erhalten bleibt (das Profile wird erst am Ende eines ISPF-Dialogs auf die Platte zurückgeschrieben), wird ein TBSAVE auf das Profile ausgegeben,

nachdem vorher noch wichtige Restart-Information gesetzt oder aus dem Shared Pool übernommen wurden. Eine Kontrollvariable im Profile-Pool signalisiert, in welchem logischen Schirm der ABEND auftrat.

- Bei normaler Beendigung einer Anwendung wird die oben erwähnte Kontrollvariable zurückgesetzt.
- Beim Wiederaufsetzen des AHF-Dialogs wird eine Recovery-Routine angezogen, in der nun verschiedene Aktionen durchgeführt werden, die einen ablauffähigen Dialog wieder garantieren sollen (siehe Abbildung 15):
 1. Setzen von AHF-Systemvariablen
 2. Die Programmumgebung wird auf die LOGON-Prozedur zurückgesetzt.
 3. Die Master-Menu-Variablen werden neu generiert.
 4. Die SYSPROC-Verkettung wird auf die Basis-Verkettung zurückgesetzt.
 5. Auflösung von LIBDEF's, sofern noch vorhanden.
 6. Freigabe sonstiger Anwendungsfiles

Geöffnete Dateien bleiben bisher jedoch geöffnet. Abhilfe ist hier nur über ein erneutes LOGON möglich. Damit der Benutzer aber die übrigen Anwendungen aufrufen kann, wird die fehlerhafte Anwendung *markiert* und beim nächsten Aufruf in der gleichen Sitzung zurückgewiesen. Beim nächsten AHF-Aufruf wird die Maskierung jedoch gelöscht, so daß die Anwendung wieder aktiviert werden kann.

```
----- APPLICATION RECOVERY INFORMATION PANEL -----  
COMMAND ==> _  
  
Wait for completion of the recovery actions displayed below:  
  
Installation names reinstated  
  
Program Environment restored to LOGON  
  
Master Panel Variables regenerated  
  
SYSPROC concatenation restored  
  
Application-related Files freed  
  
COMPLETED  
  
Press ENTER Key to continue and return to the user application Master Menu
```

Abbildung 15. In-Process-Panel nach Beendigung der Recovery

TABELLEN DER BENUTZERSCHNITTSTELLE

Allgemeine Bemerkungen zur Tabellenverarbeitung

Unter einer ISPF-Tabelle versteht man eine zweidimensionale Informationsmatrix. Jede *Spalte* enthält Werte, die mit einer bestimmten Dialogvariablen korrespondieren. Jede *Zeile* enthält einen Satz von Variablenwerten, die zusammen verarbeitet werden. Die Verarbeitung erfolgt zeilenweise.

Tabellen können temporär oder permanent gespeichert werden. Eine permanente Tabelle wird als Member in einer Tabellen-Bibliothek geführt.

Anmerkung: Die Tabellen-Datei *TABLES* wird beim ersten Aufruf der Benutzerschnittstelle automatisch angelegt.

Bei einer Referenz auf eine permanente Tabelle wird eine Kopie in den *virtuellen* Speicher übertragen, wo sie dann weiter verarbeitet werden kann, d.h. z.B. auch, daß die Tabelle am Schluß der Verarbeitung wieder zurück auf die Platte kopiert werden kann oder nicht.

Anmerkung: Die Tabellen der Benutzerschnittstelle können im *Split-Screen-Modus* bearbeitet werden.

Tabellen können dadurch aufgefüllt werden, daß Dialogvariable durch CLIST-Befehle mit Werten versorgt werden, Variable über Dateneingabe-Panels Werte zugewiesen werden, oder über ein spezielles Table-Display-Panel die Eingabe durchgeführt wird. In jedem Fall müssen die Tabellen-Variablen danach mit Befehlen des Tabellen-Service in die Tabelle gebracht werden.

Neben der Vereinbarung einer Schlüsselvariablen zur Steuerung des Zugriffs auf einzelne oder mehrere Tabellenzeilen ist es in manchen Anwendungen nützlich, bestimmte Zeilen einer vorzugebenden *Verarbeitungsvorschrift* zu unterwerfen. Beispiele hierfür sind die im ISPF/PDF verwalteten Member- oder Dateilisten.

In solche Fällen kann die erste Spalte in einer Tabellenzeile als Eingabefeld deklariert werden, wo der Benutzer dann eine Kodierung zur Beschreibung der gewünschten Aktivität eintragen kann. Ein so definierter Aktionscode ist eine Variable, die nicht zur Tabelle gehört und nicht nur in einer Zeile, sondern auch gleich vor mehrere Zeilen gesetzt werden kann.

Aufbau der Tabellen der Benutzerschnittstelle

Neben der formalen Beschreibung einer ISPF-Anwendung ist die Speicherung dieser Definitionen in ISPF-Tabellen und deren Verwaltung mit Hilfe des Dialog Management ein wesentliches Merkmal der hier vorgestellten Benutzerschnittstelle. Auch aus Performance-Gründen werden bisher die formalen Beschreibungen einer Anwendung und die Definitionen der zugeordneten Dateien in voneinander getrennten Tabellen geführt, wobei je nach Art der Anwendung die letztere nicht gelesen werden muß Grundsätzlich werden also zwei (relationale) Tabellen zur Beschreibung einer Anwendung herangezogen (siehe unten):

1. Anwendungs-Referenz-Tabelle
2. Zugeordnete Datei-Tabelle

Insgesamt werden vom AHF die folgenden Tabellen(paare) in der Anwendungsverwaltung eingesetzt:

- Aktuelle Anwendungstabelle zur Aktivierung des Master Menu
- Anwendungsbezogene Benutzer-Datei-Tabelle (wahlweise)
- Tabelle zur Verwaltung privater Anwendungen
- Speicherung der öffentlichen Anwendungen

Die folgenden Darstellungen zeigen schematisch, wie eine Anwendung innerhalb der Tabellen definiert ist. Als *Schlüssel* wird dabei die Kurzbeschreibung (Funktion) einer Anwendung benutzt.

Anwendungstabelle

Funktion	Auswahl- option	Beschreibung	Aufruf	Status Pointer	Progr.- Envir.	Auth. Code
----------	--------------------	--------------	--------	-------------------	-------------------	---------------

Der *Status-Pointer* beschreibt die Datei-Zuordnungen einer Anwendung und kann die folgenden Werte annehmen:

- 0 Keine ISPF-Dateien oder nur "öffentliche" Basis-Dateien
- 1 Es ist eine Zuordnung von Anwendungsdateien mit dem LIBDEF-Service durchzuführen. Dies ist der Normalfall.
- 2 Zusätzlich zum LIBDEF-Service (*und nur dann*) ist eine Allokation von anwendungsbezogenen Benutzerdateien durchzuführen.

Eine Funktion (Anwendung) kann demnach *keine, einen* oder *mehrere* Einträge in den unten beschriebenen Tabellen haben. Gegebenenfalls werden dadurch unnötige I/O-Vorgänge vermieden. Die ISPF-Filenamen sind in Abbildung 19 auf Seite 32 aufgeführt.

In besonderen Fällen kann über die Angabe eines *Programm-Environments* die LOGON-Prozedur umgesetzt werden (siehe "Hinzufügen einer Anwendung" auf Seite 16). In der Tabelle wird außerdem eine Variable geführt, die anzeigt, ob es sich um eine *authorisierte* Anwendung handelt, oder ob die Anwendung allgemein zur Verfügung steht.

Dateitabelle

Funktion	ISPF-Filename1	Datei-Typ1	Dateiname1
Funktion	ISPF-Filename2	Datei-Typ2	Dateiname2
Funktion	ISPF-Filename3	Datei-Typ3	Dateiname3
.....			
.....			

Benutzerdateitabelle

Funktion	ISPF-Benutzer-Filename1	Datei-Typ1	Benutzer-Dateiname1
Funktion	ISPF-Benutzer-Filename2	Datei-Typ2	Benutzer-Dateiname2
Funktion	ISPF-Benutzer-Filename3	Datei-Typ3	Benutzer-Dateiname3
.....			
.....			

Der *Dateityp* ergibt sich aus der Eigenschaft der betreffenden ISPF-Datei und wird automatisch auf die Werte P, C, M, S, F, L, A, I oder O gesetzt. I steht dabei für eine Eingabe-Tabellendatei und O für eine Ausgabe-Tabellendatei.

Der Dateityp A ist nur in Ausnahmefällen definiert und beschreibt eine allgemeine, Nicht-ISPF-Datei, die über das ALLOCATE-Kommando einer bestimmten Anwendung zuzuordnen ist. Eine solche Zuordnung wird bisher nicht unterstützt; sie sollte daher besser in eine der eigentlichen Anwendung vorausgehenden Aufruf-CLIST verlagert werden. Dies hat außerdem den Vorteil, daß bei Produktänderungen nicht auch noch die betreffenden Tabelleneinträge geändert werden müssen, wenn solche Änderungen nicht die ISPF-Umgebung, sondern andere Funktionsdateien einer Anwendung betreffen.

```
----- PUBLIC APPLICATION POOL ----- ROW 1 OF 13  
COMMAND ==> _ SCROLL ==> PAGE
```

This table shows the current predefined public ISPF applications which may be collected into your own application table

OPTION	FUNCTION	DESCRIPTION
A	APL2	A P L 2
C	RACF	Resource Access Control Facility Services
D	GML	Formatting with GML/DCF
E	EARN	European Academic and Research Network Services
F	F/VS	VS FORTRAN Services
G	GDDM	Interactive Graphic Application Services
I	INFO	Information
K	GKS	Graphisches Kernsystem
L	LISTA	Display TSO Allocation List (LISTA ST)
M	ACRITH	Online Training Component (OTC)
R	RMFMON	Systemmonitor RMFMON
T	TOOLS	FORTRAN Tools
V	F/VP	Siemens FORTRAN Scalar and Vector Services

```
***** BOTTOM OF DATA *****
```

Abbildung 16. Öffentlicher Pool für ISPF-Anwendungen

Installation der öffentlichen Anwendungs-Tabellen

Der öffentliche Anwendungspool wird mit Hilfe der Option 7.9 aus dem Hauptpanel heraus eingerichtet (siehe Abbildung 18 auf Seite 28). Diese Aktion kann jedoch nur von hierzu eigens autorisierten Benutzern während der Systemwartung durchgeführt werden. Wie bereits in "Initialisierung" auf Seite 13 beschrieben, wird ein vordefinierter Satz von Anwendungen (siehe Abbildung 16 auf Seite 26) in diese allgemein zugängliche Tabelle gestellt, die in der ISPF-Basis-Tabellendatei geführt werden kann.

Speicherung und Sicherung von Anwendungsdefinitionen

Da der Benutzer nur eine *begrenzte* Anzahl von Anwendungen in der aktuellen Anwendungstabelle, aus der das Master Menu generiert wird, unterbringen kann, muß der Benutzer die Möglichkeit haben, einen *eigenen Pool* zu verwalten. Diese Pool-Tabellen werden bei jeder Initialisierung automatisch angelegt und aus den öffentlichen Tabellen gefüllt. Anwendungen können aus der aktuellen Anwendungstabelle in diesen Pool übertragen, oder aus diesem Benutzer-Pool in die aktuelle Anwendungstabelle kopiert werden. Existiert eine Anwendung bereits in der Ziel-tabelle, so kann der Benutzer entscheiden, ob die betreffende Anwendung überschrieben werden soll, oder nicht.

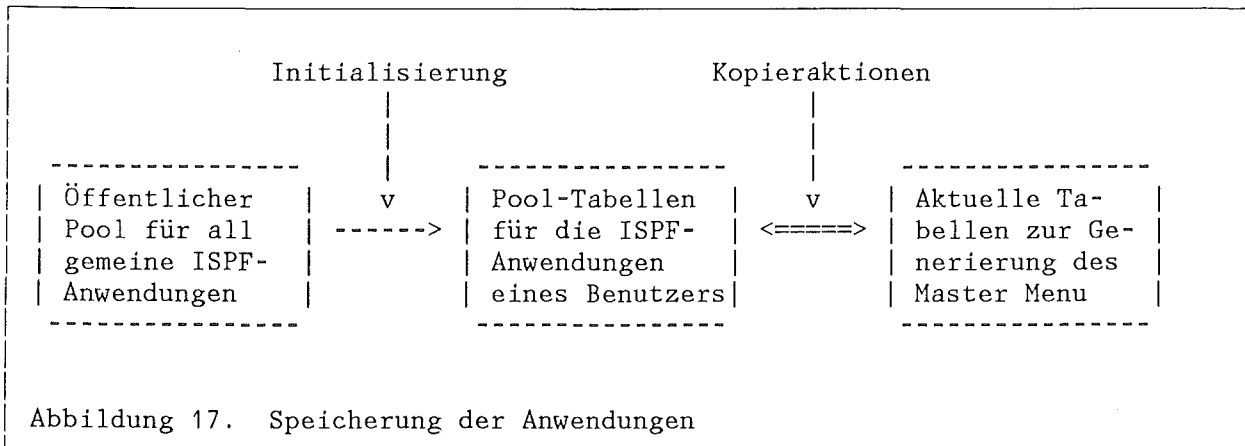


Abbildung 17 zeigt schematisch die Beziehungen zwischen den Pool-Tabellen der Benutzerschnittstelle. Sie zeigt, daß Anwendungen zwischen der aktuellen Tabelle und dem Benutzer-Pool ausgetauscht werden können. Da aus dem öffentlichen Pool Anwendungen in die eigene Tabelle automatisch bzw. gezielt übertragen werden können, kann so eine Anwendung bei Bedarf einem größeren Benutzerkreis zur Verfügung gestellt werden. Eine Kopiermöglichkeit in den öffentlichen Pool ist, wie schon oben ausgeführt, aus wartungstechnischen Gründen nicht zugelassen.

Die oben erwähnten *Dienste zur Verwaltung der Pool-Tabellen* können z.T. aus dem Tabellen-Display der Anwendungstabelle heraus und über das in Abbildung 18 auf Seite 28 gezeigte Panel (Option 7 des Hauptpanels) vorgenommen werden:

- Kopieren einer Definition in den Benutzer-Pool
- Löschen einer Definition im Benutzer-Pool

- Übertragen einer Definition aus dem Benutzer-Pool in die Anwendungstabelle (Master Menu)
- Übertragen einer Definition aus dem öffentlichen Pool in die Anwendungstabelle (Master Menu)
- Kopieren einer Definition in den öffentlichen Pool
- Löschen einer Definition im öffentlichen Pool
- Hinzufügen einer Definition im öffentlichen Pool
- Modifikation einer Definition im öffentlichen Pool
- Initialisierung des öffentlichen Pools

```
----- APPLICATION POOL MANAGEMENT SERVICES -----  
OPTION ==> _  
  
Select one of the options below:  
  
END USER POOL FUNCTIONS:  
  
1 - Save current user applications into the user pool table  
2 - Delete one or more applications in the user pool table  
3 - Copy an application from the user pool to the user application table  
4 - Copy an application from the public pool to the user application table  
  
ADMINISTRATOR FUNCTIONS FOR THE PUBLIC POOL: ( for authorised users only )  
  
5 - Save current user applications into the public pool table  
6 - Delete one or more applications in the public pool table  
7 - Define a new application in the public application pool  
8 - Modify a public application definition  
9 - Initialise public application pool from a predefined set  
  
Press END Key (PF3) to return to the APPLICATION SERVICES Panel  
  
Abbildung 18. AHF-Pool-Verwaltung: Die öffentlichen Tabellen (geschützt)  
können nicht allgemein bearbeitet werden.
```

ZUSAMMENFASSUNG UND AUSBLICK

Die hier vorgeschlagene Implementierung einer Benutzerschnittstelle zu ISPF-Anwendungen kann zur Bereitstellung von Software-Werkzeugen mit breitgefächertem Anwendungsspektrum dienen. Zum Schluß seien noch einmal einige wichtige Aspekte der Realisierung zusammengefaßt:

AHF ist ein interaktives Werkzeug zur Verwaltung von ISPF-Anwendungen.

AHF bietet insbesondere dem End-Benutzer die Möglichkeit, in einfacher Weise die von ihm benötigten Software-Werkzeuge aus öffentlichen Anwendungen zusammenzustellen und auszuführen.

- AHF* ist ein Dialogsystem für den Entwickler, neue interaktive Anwendungen formal zu beschreiben und zu testen.
- AHF* ist in der Kommandosprache CLIST geschrieben und modular aufgebaut. Die dadurch erreichte Transparenz führt trotzdem zu keinen erheblichen Performance-Einbußen, da ab TSO/E Release 2.1 spürbare Laufzeitverbesserungen von CLIST festgestellt werden konnten.
- AHF* verwendet zur Speicherung der Komponenten einer Anwendungsdefinition (Aufruf und zugeordnete Dateien für die Dialogelemente) ISPF-Tabellen.
- AHF* kann als eine unabhängige, flexible Benutzeroberfläche zu ISPF-Anwendungen angesehen werden.
- AHF* wurde unter dem Gesichtspunkt der Allgemeinheit, Benutzbarkeit und Wartbarkeit entwickelt.
- AHF* stellt hinsichtlich seiner inneren Struktur, die im wesentlichen durch eine Tabellen-gestützte Dialogfunktion charakterisiert werden kann, ein offenes System dar.

Da auch ein generiertes Master Menu nur eine feste Anzahl von Anwendungen maximal aufnehmen kann, ist der Benutzer im Bedarfsfall gezwungen, die eine oder andere Anwendung auszulagern, wenn er sich eine neue Anwendung zuordnen möchte. Da davon ausgegangen werden kann, daß die Zahl der bereitgestellten Anwendungen noch stark ansteigen wird, ist nicht nur aus sachlichen Gründen eine Erweiterung des AHF dahingehend wünschenswert, eine *Gruppierung von Anwendungen* zuzulassen. In einer solchen Situation wird dann bei Anwahl einer Gruppe ein Untermenü generiert, aus dem heraus wie in der aktuellen Version die einzelnen Anwendungen selektiert und aktiviert werden können. Die Gruppenelemente könnten in einer eigenen Tabelle gespeichert und vom Benutzer verwaltet werden; er kann dann gegebenenfalls entscheiden, welche Anwendungen er unter einer bestimmten Gruppe aufrufen möchte.

ANHANG A: AUFRUF VON ISPF-ANWENDUNGEN OHNE BENUTZER-INTERFACE

1. LOGON Prozedur

- User Profile
- Programmumgebung (SYSLIB-Verkettung)
- Pre-Allokierte Arbeitsdateien
- Dialogelemente für Produktionsumgebung (Panels, CLIST, ..)
 - ISPF/PDF System Dateien
 - KFK System Dateien

2. ISPF Start-Prozedur

- Dialogelemente für Produktionsumgebung (Panels, CLIST, ..)
 - ISPF/PDF System Dateien
 - KFK System Dateien
 - Anwendungsdateien
 - Anwendungstabellen

ANHANG B: VERTEILUNG DER ZUORDNUNGEN MIT BENUTZER-INTERFACE

1. LOGON Prozedur

- User Profile
- Pre-Allokierte Arbeitsdateien
- Dialogelemente für Produktionsumgebung (Panels, CLIST, ..)
 - ISPF/PDF System Dateien
 - KFK System Dateien

2. Benutzer-Schnittstelle AHF (ISPF Start-Prozedur)

- Programmumgebung (SYSLIB-Verkettung)
- *Aktivierung von Anwendungen*
 - Dynamische Anpassung der Programmumgebung durch die Anwendung
 - Dynamische Zuordnung der Anwendungsdateien
 - Aufruf der Anwendung

ANHANG C: START-CLIST FÜR DAS BENUTZER-INTERFACE

```
PROC 0 TEST TRACE TESTX TRACEX OPT() 000000
/*****/ 000000
/*      I S P F  -  B E N U T Z E R  -  I N T E R F A C E      */ 000000
/*****/ 000000
/*  START-CLIST DES APPLIKATIONS/DATEI-VERWALTUNGSSYSTEMS VERSION 2  */ 000000
/*****/ 000000
/*****/ 000000
CONTROL END(ENDO) NOMSG NOFLUSH MAIN 000000
GLOBAL AHSVTIO1,AHSVTIO2,AHSVTIO3,AHSVTIO4,AHSVTIO5 000000
GLOBAL AHSVTOO1,AHSVTOO2,AHSVTOO3,AHSVTOO4,AHSVTOO5 000000
IF &OPT= THEN SET &O= 000000
      ELSE SET &O= CMD(&OPT) 000000
/*****/ 000000
/*  PRUEFEN, OB DIE TABELLEN-DATEI BEREITS ANGELEGT WURDE, ODER NICHT  */ 000000
/*****/ 000000
IF &SYSDSN('&SYSPREF..TABLES') ^= OK THEN + 000000
  DO 000000
    FREE ATTR(FB80) 000000
    ATTR FB80 BLKSIZE(3120) LRECL(80) RECFM(F B) 000000
    ALLOC DA('&SYSPREF..TABLES') + 000000
      SPACE(5 2) TRACKS NEW USING(FB80) DIR(5) 000000
    FREE ATTR(FB80) 000000
  ENDO 000000
/*****/ 000000
/*  BASIS-VERKETTUNG VON ISPTLIB/ISPTABL BESTIMMEN UND ALLOCATE      */ 000000
/*****/ 000000
%AHSCBBV 000000
/*****/ 000000
/*  AUFRUF DES HAUPTMENUES ---> PANEL AHS@MSTR UEBER AHSCMAIN      */ 000000
/*****/ 000000
/*****/ 000000
/*****/ 000000
ISPSTART CMD(%AHSCMAIN &O) NEWAPPL(AHS) &TEST &TRACE &TRACEX &TESTX 000000
000000
/*****/ 000000
/*****/ 000000
/*  BASIS-VERKETTUNG VON ISPTLIB/ISPTABL ZURUECKSETZEN      */ 000000
/*****/ 000000
%AHSCBRB 000000
/*****/ 000000
/*  */ 000000
/*****/ 000000
/*  ENDE DES ISPF-BENUTZER-INTERFACE      */ 000000
/*****/ 000000
END 000000
```

ANHANG D: ISPF-DIALOGBAUSTEINE

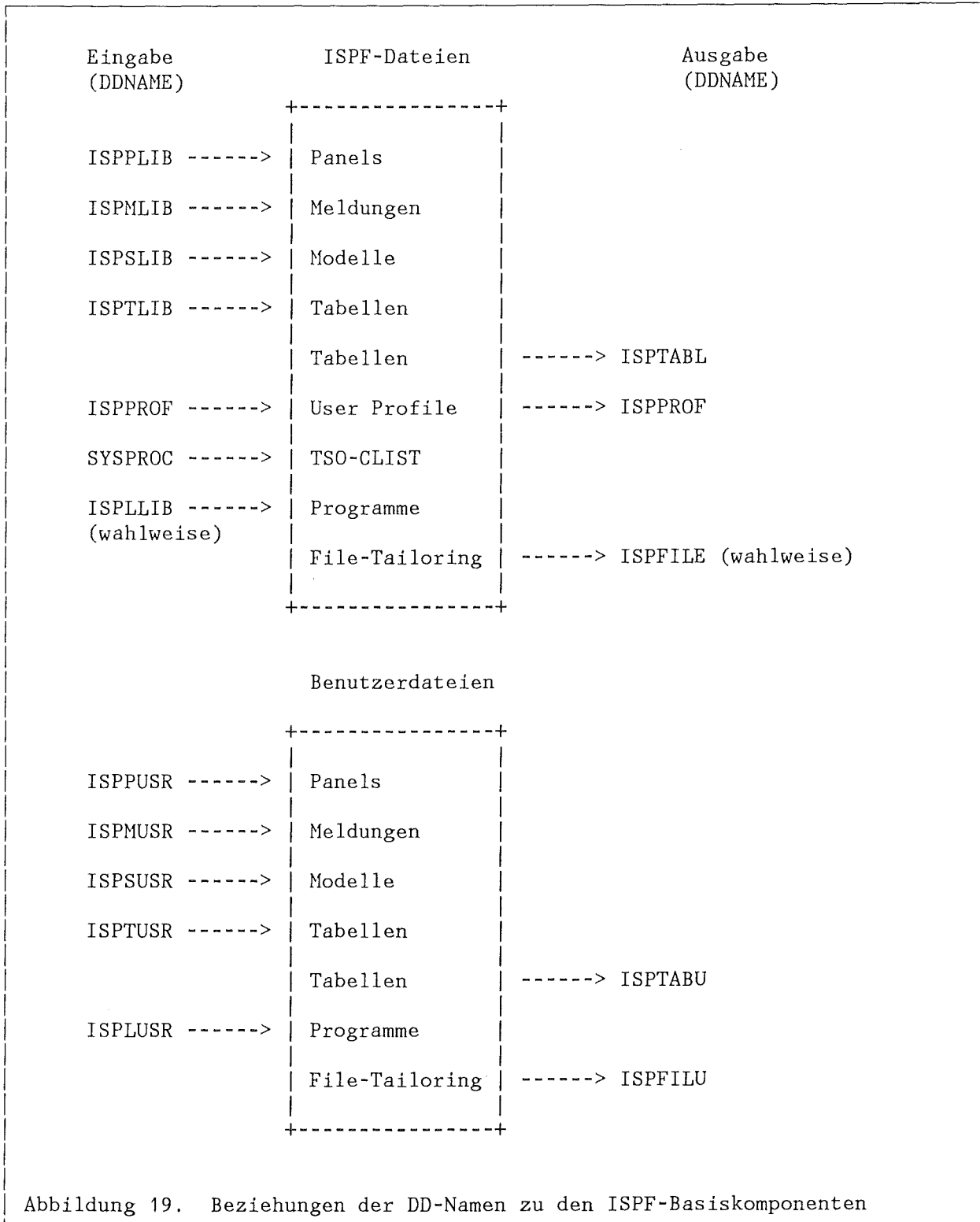


Abbildung 19. Beziehungen der DD-Namen zu den ISPF-Basiskomponenten

ANHANG E: DYNAMISCHE ZUORDNUNG VON ISPF-DATEIEN

	LIBDEF/DATASET	LIBDEF/LIBRARY	kein LIBDEF
Panels	ISPPUSR	LIBDEF	ISPPLIB
	LIBDEF	ISPPLIB	
	ISPPLIB		
Messages	ISPMUSR	LIBDEF	ISPMLIB
	LIBDEF	ISPMLIB	
	ISPMLIB		
Eingabetabelle	ISPTUSR	LIBDEF	ISPTLIB
	LIBDEF	ISPTLIB	
	ISPTLIB		
Ausgabetabelle	ISPTABU	LIBDEF	ISPTABL
	LIBDEF		
Skeletons	ISPSUSR	LIBDEF	ISPSLIB
	LIBDEF	ISPSLIB	
	ISPSLIB		
File Tailoring	ISPFILU	LIBDEF	ISPFILU
	LIBDEF		
Linklib	JOB PACK AREA	JOB PACK AREA	JOB PACK AREA
	ISPLUSR		ISPLLIB
	LIBDEF	LIBDEF	STEPLIB
	LINK PACK AREA	LINK PACK AREA	LINK PACK AREA
	LINKLIB	LINKLIB	LINKLIB
	JOB PACK AREA	JOB PACK AREA	
	ISPLLIB	ISPLLIB	
	STEPLIB	STEPLIB	
	LINK PACK AREA	LINK PACK AREA	
	LINKLIB	LINKLIB	
Eingabetabelle (mit LIBRARY)	allokierte Datei LIBDEF	LIBDEF allokierte Datei	allokierte Datei
Ausgabetabelle (mit LIBRARY)	allokierte Datei LIBDEF	LIBDEF	allokierte Datei
File Tailoring (mit LIBRARY)	allokierte Datei LIBDEF	LIBDEF	allokierte Datei

Abbildung 20. Suchreihenfolge für ISPF-Dateien bei LIBDEF-Service