

KfK 2644  
Juni 1978

# Grundlagen und Funktionen von Report- Generatoren

F.-J. Polster, D. Stöckle  
Institut für Datenverarbeitung in der Technik

Kernforschungszentrum Karlsruhe

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

KERNFORSCHUNGSZENTRUM KARLSRUHE GMBH

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Datenverarbeitung in der Technik

KfK 2644

GRUNDLAGEN UND FUNKTIONEN VON REPORT-GENERATOREN

F.-J. Polster

D. Stöckle

Kernforschungszentrum Karlsruhe GmbH, K a r l s r u h e



## Zusammenfassung:

Auf der Grundlage einer Literaturübersicht werden die Möglichkeiten und Eigenschaften von Report-Generatoren zur Erzeugung von Berichten dargestellt und exemplarisch an existierenden Systemen verdeutlicht.

Die Begriffe zur Funktionsweise eines Report-Generators und zum Aufbau eines Berichts werden einheitlich definiert. Die Funktionen der Report-Definition, -Modifikation, Datenauswahl, Datenverknüpfung, Ausgabesteuerung und Zugriffsschutz sind die Hauptbestandteile eines Report-Generators und werden erläutert. Zusätzlich werden die Schnittstellen des Report-Generators als Bestandteil eines Informationssystems erklärt.

Es werden nicht nur Merkmale existierender Systeme genannt, sondern auch Konzepte und Ideen, die noch nicht in einem bestehenden System implementiert sind.

## TERMINOLOGY AND FUNCTIONS OF REPORT-GENERATORS

### Abstract:

Based on a survey of literature the features and facilities of report-generators are presented; existing operational systems are used as examples. A uniform terminology is introduced to describe the functions of a report-generator and the structure of a report. The main functions are: report definition, report modification, data selection, data evaluation, output generation, access control. The interfaces of a report-generator as a component of information systems are explained. In addition to the characteristics of operational systems concepts given solely in literature are presented.

## Inhalt

	<u>Seite</u>
1. Einleitung	1
2. Funktionen eines Report-Generators	3
2.1 Report-Definition	3
2.1.1 Elemente einer Report-Definitionssprache	5
2.1.2 Report-Strukturanweisungen	6
2.1.3 Report-Formatierung	10
2.1.4 Sortieranweisungen	11
2.1.5 Bedingte Ausführung von Anweisungen	12
2.2 Report-Modifikation	13
2.3 Datenauswahl	14
2.4 Datenverknüpfung	15
2.4.1 Grundoperationen	17
2.4.2 Komplexe Funktionen	17
2.5 Ausgabesteuerung	18
2.6 Zugriffsschutz	22
3. Schnittstellen	23
3.1 Schnittstelle zum Dialogsystem	24
3.2 Schnittstelle zum Datenverwaltungssystem	26
3.3 Schnittstelle zur problemorientierten Software	29
3.4 Schnittstelle zum Betriebssystem	30
4. Ergebnisse und Ausblick	31
5. Literatur	32

## 1. Einleitung

Eine wesentliche Funktion eines Informationssystems ist das Wiederauffinden bestimmter Daten aus einer Datenbasis und deren Aufbereitung zur Ausgabe in einer benutzergerechten, verständlichen Form. Das Ergebnis einer solchen Auswertung wird im folgenden "Bericht" oder "Report" genannt.

Die allgemeinen Programmiersprachen sind nicht auf diese Aufgabenstellung zugeschnitten und bieten kaum Hilfsmittel (Datenstrukturen, Operationen) zur einfachen Erstellung eines Berichts.

Zu diesem Zweck wurden daher sogenannte "Report-Generatoren" entwickelt. Diese Systeme sollen den Programmieraufwand zur Erzeugung eines Berichts klein halten und damit zur Reduktion der Software-Kosten beitragen; zugleich sollen sie ein Werkzeug für die Gruppe der "anspruchsvollen Laien" /WEDH 74/ ('analysts and researchers' /CODE 74/), wie z.B. Fachleute verschiedener Disziplinen, darstellen:

Der Benutzer hat auf leicht erlernbare, verständliche Weise festzulegen, welche Daten ausgewertet werden sollen und wie die äußere Form des Berichtes aussehen soll. Der Report-Generator übernimmt das Daten-Retrieval, die Aufbereitung und gegebenenfalls Verknüpfung der Daten. Die Ausgabe des Berichtes kann auf verschiedenen Geräten erfolgen.

Diese Arbeit hat folgende Ziele:

- 1) Es soll eine Übersicht über die wesentlichen Funktionen von existierenden Systemen zur Report-Erstellung gegeben werden. Hierzu wurden untersucht:
  - FOIN (Formular Interpreter) /FOIN/ ist Bestandteil der Sprache MAGICS für das Kleinrechnersystem DIETZ-621. Es ermöglicht die Beschreibung von Ein- und Ausgabe-Formularen.
  - IMS (Information Management System) /CAGE 76/ ein Datenbanksystem mit Report-Generator des Mathematics and Computation Laboratory, Federal

Preparedness Agency (Washington, D.C.), für die Rechnerserie UNIVAC 1100 (nicht zu verwechseln mit IMS von IBM!).

- IREG (Interaktiver Report-Generator) /IREG 77/ wurde von der Gesellschaft für Mathematik und Datenverarbeitung (GMD) als Baustein des Formularorientierten Interaktiven Datenbanksystems (FIDAS) entwickelt und läuft auf den Großrechnern SIEMENS 4004 und 7.700 .
- PARLIS (Parametergesteuertes System zur Erzeugung von Listen und Formularen) /PARL/ ist ein Report-Generator des Mathematischen Beratungs- und Programmierungsdienstes (mbp) für Kleinrechner in der kommerziellen Anwendung.
- QUERY/3000 von Hewlett-Packard /QUER 76/ ist die Anfragesprache für das Datenverwaltungssystem IMAGE/3000 und enthält einen Satz von Anweisungen zur Erzeugung von Reports.
- SYSTEM 2000, ein Datenbanksystem mit Report-Generator der MRI-Systems Corporation (Austin/Texas) /SYST 74/, für Großrechner der Firmen CDC, IBM und UNIVAC .

Es ist jedoch nicht beabsichtigt, diese Systeme in allen Einzelheiten darzustellen oder gar einem Vergleich zu unterziehen. Spezielle Fähigkeiten einzelner Systeme werden nur dann genannt, wenn sie besonders wesentlich erscheinen.

2) Die Ergebnisse eines zu diesem Thema durchgeführten allgemeinen Literaturstudiums sollen dargestellt werden; dies betrifft also neue Konzepte oder Verallgemeinerungen bereits implementierter Funktionen, ebenso wie allgemeine Fragen der Schnittstellen eines Report-Generators.

3) Da in der Literatur keine Einheitlichkeit in der Bezeichnungsweise herrscht, sollen vor allem im Hinblick auf die weiteren Arbeiten zum Thema "Report-Generator" die wesentlichen Begriffe geklärt und der strukturelle Aufbau eines Berichts definiert werden.



## 2. Funktionen eines Report-Generators

Ein Report-Generator setzt sich aus folgenden Funktionen zusammen (vgl. Bild 1):

- Report-Definition und -Modifikation  
Festlegung bzw. Änderung von Aufbau, Inhalt und Format eines Berichts, d.h. des Report-Programms
- Datenauswahl  
Selektion und Extraktion der gewünschten Berichtsdaten aus der Datenbasis
- Datenverknüpfung  
Durchführung von Berechnungen zur (anwendungsspezifischen) Auswertung
- Ausgabesteuerung  
Abbildung des logischen Berichts auf das physikalische Ausgabemedium
- Zugriffsschutz
  - . Verhinderung des unberechtigten Aufrufs von Funktionen zur Report-Definition und -Modifikation (Bedienungsschutz)
  - . Verhinderung des unberechtigten Zugriffs auf Daten der Datenbasis oder abgeleiteter Informationen (Datenschutz).

### 2.1 Report-Definition

Unter Report-Definition wird hier ausschließlich die Beschreibung des Aufbaus und der Form eines Berichts verstanden; die Spezifikation der Daten der Datenbasis, die für den Bericht auszuwerten sind, ist Gegenstand der Funktion Datenauswahl (Abschnitt 2.3).

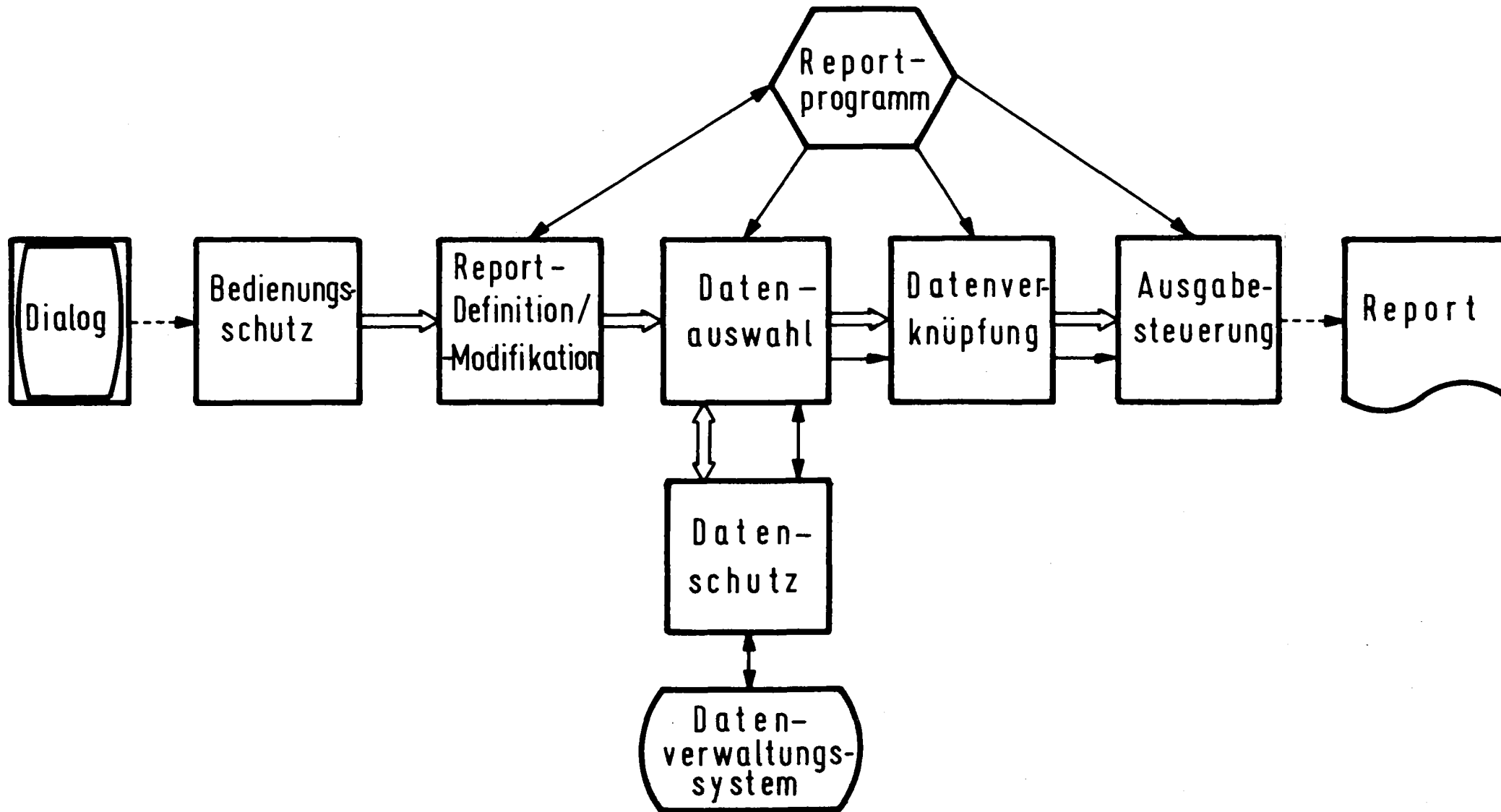


Bild 1: Funktionen eines Report - Generators  
 (⇒≙Kontrollfluß; →≙ Datenfluß)

Eine Report-Definition kann als Programm einer Report-Definitionssprache ("Report Definition Language", RDL) aufgefaßt werden. Die Konzepte und Operationen zur Report-Definition werden in den nächsten Abschnitten unter diesem Aspekt beschrieben.

### 2.1.1 Elemente einer Report-Definitionssprache (RDL)

Als Objekte einer RDL stehen (explizit oder implizit) zur Verfügung:

- Attribut
- Report-Record
- Report-Variable, Report-Konstante
- Report-Zeile
- Report-Seite

Zur Spezifikation der Daten, die für einen Bericht benötigt werden, bedient man sich zur Report-Definition der Attribute (oder genauer Attributnamen): Sie bezeichnen die elementaren, nicht weiter zerlegbaren Objekte der Datenbasis; ein anderer, häufig benutzter Begriff hierfür ist "Feld" (hierarchisches Datenmodell) oder "item" (CODASYL).

Die Attribute einer Report-Definition definieren den Typ eines Report-Records; ein Report-Record, also die Ausprägung eines Report-Recordtyps, enthält die aus dem entsprechenden Strukturelement der Datenbasis extrahierten Werte der Attribute des Report-Recordtyps (vgl. Abschnitt 2.3).

Die Definition von "entsprechendes Strukturelement" hängt vom Datenmodell der Datenbasis ab:

Bei Verwendung eines relationalen Datenmodells zum Beispiel entspricht einem Report-Record im wesentlichen das "Tupel"; komplizierter ist die Erklärung für ein hierarchisches Datenmodell; hier sind die Pfade von der Wurzel bis zu den Blättern der Baumstruktur zu betrachten.

Report-Variablen entsprechen den Variablen bei Programmiersprachen. Sie erlauben das Zwischenspeichern von Attributwerten und können als Zähler, für Arithmetik (siehe Abschnitt 2.4) u.a. benutzt werden. Als Report-Variablentypen sind daher alle Attributtypen der Datenbasis vorgesehen (z.B. numerisch, string). Zusätzlich findet man für spezielle Anwendungen vor allem im kommerziellen Bereich weitere Variablentypen wie MONEY, DATE, COUNTER. Für Report-Konstante gilt entsprechendes: sie werden als string-Konstanten - etwa zur Spezifikation von Überschriften - oder als numerische Konstante zur Formulierung von arithmetischen Operationen gebraucht.

Ein Report besteht aus (logischen) Report-Seiten, diese wiederum aus (logischen) Report-Zeilen. Eine Report-Zeile setzt sich aus den Attributwerten eines oder mehrerer Report-Records, berechneten Werten u.a. zusammen (siehe Abschnitt 2.1.3). Wesentliche Funktion einer Report-Definition ist die Beschreibung des Inhalts von Report-Zeilen und -Seiten; dies wird im Abschnitt 2.1.2 behandelt.

### 2.1.2 Report-Strukturanweisungen

(Report-)Strukturanweisungen einer RDL erlauben es, den Bericht als solchen wie auch die Seiten des Berichts zu gliedern.

#### a) Struktur eines Berichts

Für einen Bericht lassen sich 3 Teile spezifizieren (vgl. Bild 2):

- der Report-Kopf bildet die Einleitung des Berichts und kann sich über mehrere Seiten erstrecken. Im allgemeinen ist dieser Teil unabhängig von den Report-Records. Eine Titelseite ist ein Beispiel für einen Report-Kopf.
- der Report-Rumpf folgt als zweiter Teil des Berichts auf den Report-Kopf. Der Report-Rumpf enthält die Daten der Report-Records, eventuell nach Verknüpfung und Umformung.

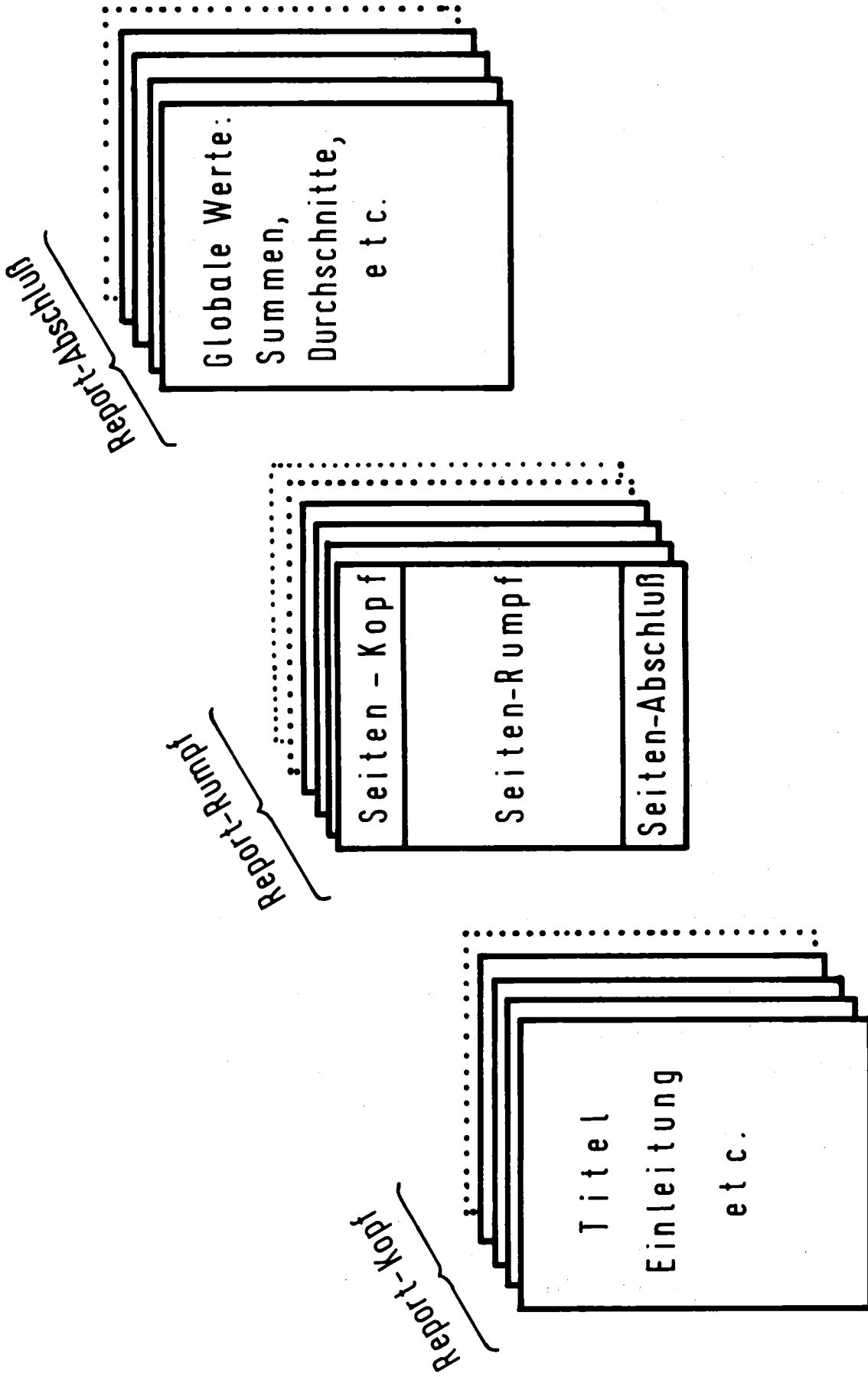


Bild 2: Struktur eines Berichts

Die Definition von (Report-Record-) Gruppen ist ein Mittel zur Strukturierung des Report-Rumpfes: Es werden ein oder mehrere Attribute als "Gruppenattribute" spezifiziert; Report-Records, die im ersten Attribut übereinstimmen, bilden eine Gruppe. Sind mehrere Gruppenattribute angegeben, so werden die aus der Gruppierung nach dem ersten Gruppenattribut resultierenden Gruppen in Untergruppen mit dem zweiten Gruppenattribut als Gruppierungskriterium unterteilt, usw. (Gruppen-Schachtelung). Um diese Gruppierung bei der Ausgabe des Berichts deutlich werden zu lassen, werden die Werte der Gruppenattribute nur beim ersten Report-Record der (Unter-)Gruppe ausgegeben, und in den folgenden Zeilen der (Unter-)Gruppe unterdrückt (vgl. Bild 3).

- der Report-Abschluß bildet den letzten Teil eines Berichts. Hier können Aussagen über alle Report-Records und/oder den Bericht insgesamt gemacht werden: Durchschnitte, Gesamtsummen, Umfang des Berichts etc.

#### b) Struktur einer Seite

Entsprechend bestehen die Seiten des Berichtsrumpfs aus

- einem Seiten-Kopf, der bei Beginn einer neuen Seite ausgegeben wird
- einem Seiten-Rumpf
- einem Seiten-Abschluß, der am Ende jeder Seite ausgegeben wird.

Ganz analog zum Berichtsrumpf ergibt sich der Seitenrumpf unmittelbar aus der Verarbeitung der Report-Records, während der Seitenkopf und der Seitenabschluß mehr globale Informationen über die Seite oder die bis dahin gelesenen Report-Records enthalten, wie z.B. Seitennummer, Summen, Durchschnittswerte.

Es besteht die Möglichkeit, die Seitengröße zu bestimmen; dies geschieht durch Angabe der Seitenlänge (Anzahl der Zeilen pro Seite) und der Seitenbreite (Zeilenlänge).

1.Gruppen- attribut	2.Gruppen- attribut	.....
------------------------	------------------------	-------

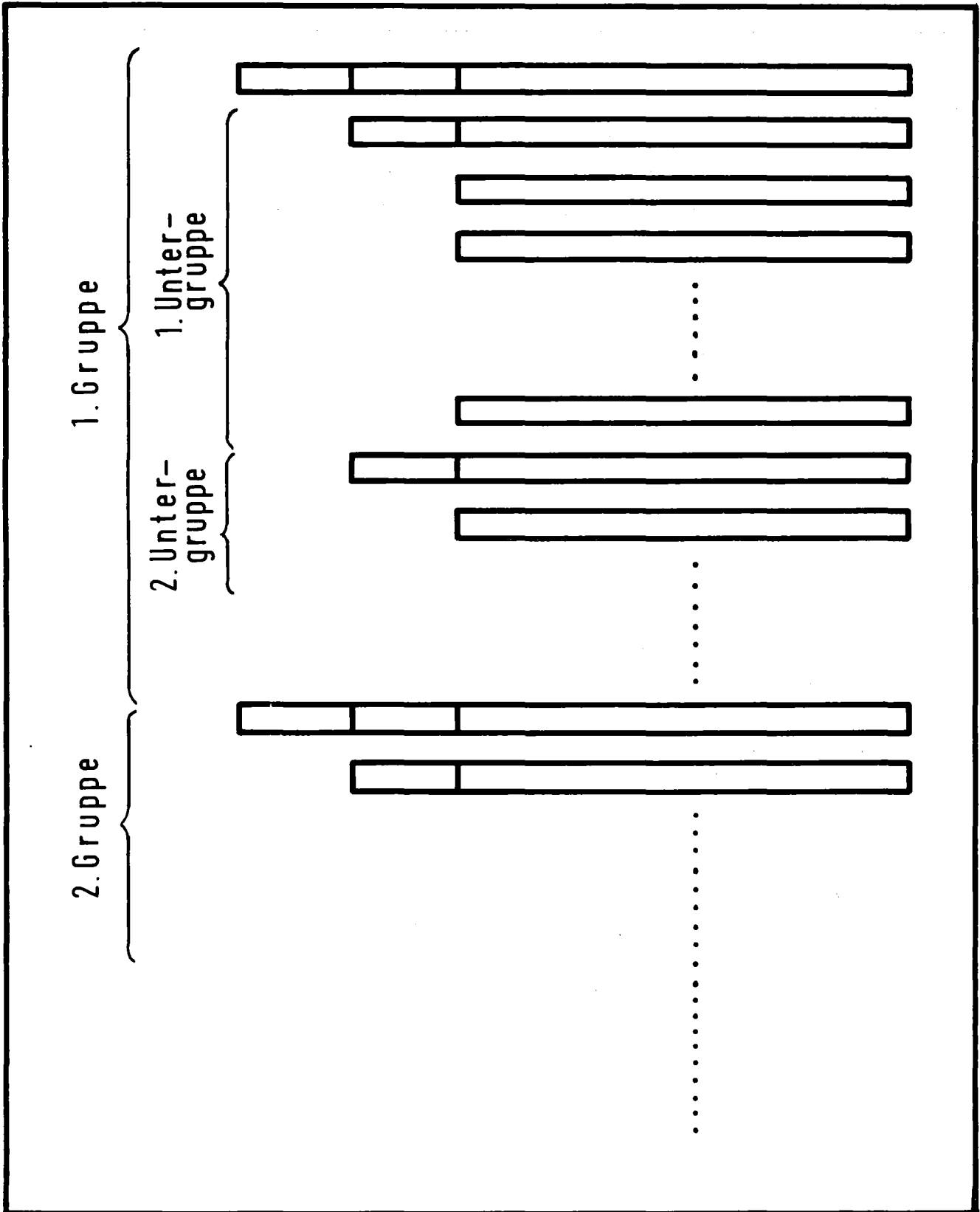


Bild 3: Gruppenbildung im Report - Rumpf

### 2.1.3 Report-Formatierung

In Analogie zu den Ausgabeanweisungen allgemeiner Programmiersprachen wird der Inhalt einer Report-Zeile definiert durch das Auflisten der gewünschten Attribute, Report-Variablen, -Konstanten und Namen von Funktionen, die einen Wert liefern (vgl. Abschnitt 2.4). Das jeweilige Ausgabeformat kann durch Angabe eines Formatcodes vom Benutzer spezifiziert werden oder ist ein i.a. typspezifisches Standardformat als Default-Formatcode.

Formatcodes stellen im wesentlichen Editiermasken dar und beschreiben die Länge und Form des auszugebenden Wertes im Bericht. Die Verwendung von "insertion characters" erlaubt das Einfügen von Zeichen wie "§", "." in die Zeichenkette.

Das folgende Beispiel (aus /SYST 74/, S. 246) soll dies illustrieren:

In den Formatcodes (in /SYST 74/ "edit picture" genannt) bedeutet:

- 9 : Zahl
- X : Buchstabe
- + : Vorzeichen ausgeben.

Als insertion characters werden in diesem Beispiel benutzt: § - . , 0

<u>auszugebender Wert</u>	<u>Formatcode</u>	<u>Ausgabe</u>
48.34	§99.99	§48.34
4834	9,999	4,834
292	+999	+292
-292	+999	-292
ABCDE	XX-X-XX	AB-C-DE
ABCDE	.XOXXXOX.	.A0BCDOE.



Ein weiteres Mittel zur Beschreibung von Report-Zeilen stellt die TABULATOR-Anweisung dar: sie erlaubt es, absolut oder relativ zur aktuellen Zeilenposition anzugeben, von welcher Stelle ab in der Zeile eine Zeichenkette abzulegen ist.

Dies kann benutzt werden, z.B. zur Spezifikation der Spalten einer Tabelle; hierfür werden auch häufig spezielle Anweisungen bereitgestellt, die eine Einteilung der Seite in Spalten erlauben. Spaltenüberschriften können explizit definiert werden; als Default-Wert ist häufig der Attributname (falls zutreffend) vorgesehen. Zur Hervorhebung der Spalteneinteilung können "Trennlinien" zwischen 2 Spalten ausgegeben werden, entsprechend kann die deutliche Trennung von Zeilen der Tabelle durch Trennzeilen gefordert werden.

Der Vollständigkeit halber seien noch die Ausgabeanweisungen

"n Leerzeilen ausgeben"

"Mit neuer Seite beginnen"

aufgeführt.

Häufig ist es angemessen und erwünscht, die Information eines Berichts in graphischer Form darzustellen; in /LEVG 74/ werden Diagramme zur Darstellung von Daten angeboten, z.B. Histogramme zur graphischen Darstellung der Häufigkeit von Werten.

#### 2.1.4 Sortieranweisungen

Sortieranweisungen werden als Mittel zur Strukturierung des Berichtsrumpfes bereitgestellt. Sie beschreiben, in welcher Reihenfolge die Report-Records durch das Report-Programm zu verarbeiten sind. Im allgemeinen können als Ordnungskriterien ein oder mehrere Attribute angegeben werden, wobei spezifisch für jedes der genannten Attribute "aufsteigend geordnet" oder "absteigend geordnet" vereinbart werden kann.

Ähnlich wie bei der Gruppenbildung werden die Report-Records zunächst nach dem ersten Attribut geordnet; die Report-Records, die im ersten Attribut übereinstimmen, werden nach dem zweiten Attribut des Ordnungskriteriums geordnet, etc.

#### 2.1.5 Bedingte Ausführung von Anweisungen

Die Ausführung von Anweisungen einer Report-Definition kann vom Vorliegen einer Bedingung abhängig gemacht werden; hierzu werden den IF-Abfragen allgemeiner Programmiersprachen ähnliche Anweisungen benutzt.

Bedingungen können allgemeine Boole'sche Ausdrücke unter Verwendung von Report-Variablen, Attributen etc. sein.

Ein wichtiger Spezialfall einer Bedingung ist der sogenannte "changes-test": Es wird überprüft, ob sich der Wert eines Attributs seit dem letzten Lesen eines Report-Records oder einer Report-Variablen seit der letzten Berechnung geändert hat; trifft dies zu, so ergibt der changes-test den Wert "TRUE", sonst "FALSE" /CAGE 76/. Hiermit läßt sich z.B. leicht erreichen, daß der Wert eines Attributs im Bericht nur dann erscheint, wenn er sich geändert hat, oder daß bei jeder Änderung eines Attributs eine Auswertung der bis dahin angefallenen Daten vorgenommen wird.

Eine Besonderheit von SYSTEM 2000 sind die "bedingten Ausgabeanweisungen"

DPRINT (detail print) und  
SPRINT (summary print)

(die neben der gewöhnlichen PRINT-Anweisung vorgesehen sind).

Eine DPRINT-Anweisung z.B. ist von der Funktion her identisch mit der PRINT-Anweisung. Im Gegensatz zu dieser kann jedoch die Ausführung einer DPRINT-Anweisung unterdrückt werden; hierzu steht die Anweisung SUPPRESS DETAIL zur Verfügung: Die nachfolgenden DPRINT-Anweisungen führen zu keiner Ausgabe, bis ein INCLUDE DETAIL das SUPPRESS DETAIL rückgängig macht.

Entsprechendes gilt für SPRINT, SUPPRESS SUMMARY und INCLUDE SUMMARY .

Solche Anweisungen werden vor allem dazu benutzt, um den Umfang und Detaillierungsgrad eines Berichts auf einfache Art zu steuern (Übersicht für's Management, Einzelheiten für Sachbearbeiter).

## 2.2 Report-Modifikation

Die Report-Modifikation hat folgende Funktionen:

### a) statische Änderung des Report-Programms

Da bei komplizierten Anwendungen des Report-Generators die erzeugte Ausgabe meistens nicht auf Anhieb den Wünschen des Benutzers entspricht, kann die im DV-System abgelegte Definition des Reports, das Report-Programm, vom Benutzer geändert werden. Der Report-Generator bietet dazu Funktionen an, die man auch in einem üblichen Editor findet. Insbesondere können Teile anderer Report-Programme in das aktuelle Report-Programm eingefügt werden.

Der Report-Generator IREG/GMD bietet dem Benutzer einen "TRACE-Modus" an, mit dessen Hilfe Anweisungen in der Report-Definition, die zu unerwünschten Ausgaben führen, leicht festgestellt werden können. Der TRACE-Modus bewirkt bei Report-Erstellung eine schrittweise Ausführung, d.h. ein Anhalten nach der Verarbeitung eines Parameters und die Anzeige des aktuellen Status und des bis dahin definierten Berichtsbildes.

### b) temporäre Änderung des Report-Programms

Es besteht die Möglichkeit, Elemente des Reports so zu definieren, daß sie erst vor Report-Erstellung durch Dialog mit ihrem aktuellen Wert belegt werden. Ebenso können vorher definierte Default-Werte vorübergehend durch aktuelle Werte ersetzt werden.

c) Editieren des erzeugten Reports

Wenn es sich nicht lohnt, wegen einer einmalig gewünschten Veränderung des generierten Reports die Report-Definition zu verändern, kann der Report vor der endgültigen Ausgabe editiert werden. Es werden Funktionen zum Versetzen, Austauschen, Löschen von Zeilen oder Teilen davon angeboten.

2.3 Datenauswahl

Wie schon in Abschnitt 2.1 erklärt, ist die Datenauswahl als eine von der Report-Definition getrennte Funktion anzusehen. Während bei der Berichts-Definition die Anordnung und das Format der Daten in einer Berichts-Zeile abstrakt festgelegt werden, erfolgt bei der Datenauswahl die Angabe, welche Daten konkret im Bericht erscheinen sollen. Durch Festlegung der Report-Records werden die entsprechenden Daten aus der Datenbasis selektiert.

Aufbau und Inhalt der Anweisungen zur Datenauswahl sind abhängig von der Schnittstelle, die das Datenverwaltungssystem für das Retrieval bietet (vgl. Abschn. 3.2). Die vom Benutzer gegebene Spezifikation der gewünschten Daten wird in eine Anfrage an das Datenverwaltungssystem umgesetzt.

Bei IMS und IREG formuliert hierzu der Benutzer eine sogenannte Auswahlbedingung. Diese dient zur Auswahl der Report-Records. Jeder Datensatz der Datenbasis wird mit Hilfe einer Vergleichsoperation auf bestimmte Attributwerte geprüft. Nur solche Datensätze werden als Report-Records im Bericht verarbeitet, deren zu vergleichende Attribute die Auswahlbedingung erfüllen (z.B.: Sätze einer Personaldatei, bei denen die Auswahlbedingung: "Dienstalter  $\geq$  5" erfüllt ist).

Wenn in einem Report-Recordtyp, wie er im Report-Programm definiert ist, nicht alle Attribute des entsprechenden Datensatzes festgelegt sind, dann

muß der Selektion eine Extraktion folgen. Die Extraktion wählt aus jedem selektierten Datensatz die im Report-Recordtyp spezifizierten Attributwerte aus (vgl. Bild 4).

Ein oder mehrere verschiedene Report-Records werden nach der Datenverknüpfung anhand des Report-Programms zu einer oder mehreren Report-Zeilen aufbereitet.

Die Report-Records werden zwischengespeichert; darauf können mehrere Berichts-Programme ausgeführt und somit verschiedene Berichte erzeugt werden /QUER 76/. Umgekehrt ist es möglich, zu einem Report-Programm aus verschiedenen Dateien Report-Records auszuwählen und so mit einem Report-Programm mehrere, inhaltlich verschiedene Berichte zu erzeugen.

## 2.4 Datenverknüpfung

Zur Verknüpfung von Report-Records nach der Datenauswahl werden numerische und nicht-numerische Funktionen zur Verfügung gestellt. Der Benutzer hat dadurch die Möglichkeit, die einmal gespeicherten Daten auf verschiedene Arten auszuwerten, ohne dafür ein selbstständiges Programm erstellen zu müssen. Der Umfang an Funktionen ist bei den untersuchten Systemen sehr unterschiedlich, beginnend bei den arithmetischen Grundrechenarten und endend bei komplexen Funktionen. Einige Systeme, z.B. SYSTEM 2000 und IREG erlauben den Anschluß benutzerdefinierter Unterprogramme zur Datenverknüpfung (vgl. Abschnitt 3.3). Bei einigen Systemen kann sich der Benutzer zur Auswertung auf Report-Variable (vgl. Abschnitt 2.1.1) stützen, zum Beispiel:

- Datentyp-spezifische Report-Variable bei QUERY ("Register") und bei IMS zur Verknüpfung der Anwenderdaten mit Hilfe der 4 Grundrechenarten.
- "E-Felder" bei IREG zur Berechnung, Zwischenspeicherung und Ausgabe numerischer Werte. Report-Parameter (z.B. akt. Zeilen- oder Spaltenzeiger) können in die Verknüpfung eingehen.

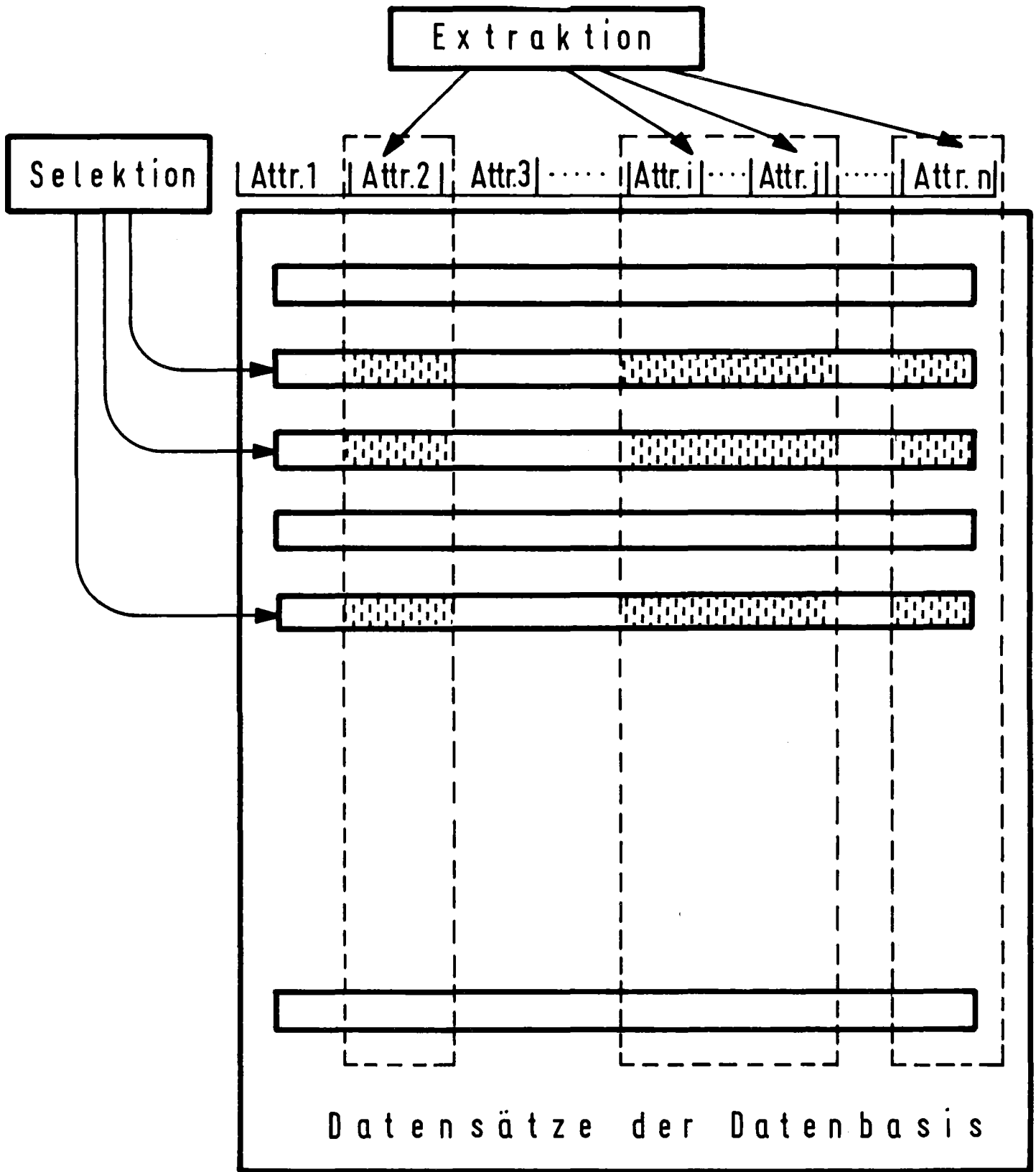



Bild 4: Schematische Darstellung der Selektion und Extraktion  
(  $\hat{=}$  extrahierte Attributwerte)

- Zählerfelder bei PARLIS  
z.B. zur Seitennumerierung

#### 2.4.1 Grundoperationen

Einfachste Systeme bieten die Verknüpfungen: Addition, Subtraktion, Multiplikation und Division an. Eine Möglichkeit zur Bildung von geschachtelten arithmetischen Ausdrücken, z.B. durch Klammerung, fehlt (PARLIS, QUERY). Daneben findet man Systeme, die arithmetische Ausdrücke wie in Standard-FORTRAN und Boole'sche Ausdrücke zulassen /LEVG 74/. Außerdem sind spezielle Funktionen vorhanden zur:

- Bestimmung der aktuellen Seitennummer
- Bestimmung des aktuellen Zeitpunktes (Datum, Uhrzeit).

Diese Funktionen stellen Werte bereit, die unabhängig vom Report-Inhalt sind, und entsprechen etwa den "built-in-functions" in FORTRAN.

#### 2.4.2 Komplexe Funktionen

Zu diesen rechnet man solche Funktionen, die aus mehreren Grundoperationen zusammengesetzt sind, und/oder mehrere Operanden haben. Man kann folgende Funktionsklassen unterscheiden:

##### a) Summenbildung

Diese Funktion läßt sich anwenden auf die Grundelemente des Berichts. Es ist möglich, die Werte eines Attributs oder einer Report-Zeile aufzusummieren. Die Attributsummen können sich auf die Attributwerte in-

nerhalb einer Gruppe, einer Seite oder des ganzen Berichts beziehen. Ebenso können die Zeilensummen einer Gruppe, einer Seite oder des Berichts zu einer Gesamtsumme addiert werden.

b) statistische Funktionen

Einfachste Funktion dabei ist die Zählung z.B. von Report-Zeilen oder unterschiedlichen Attributwerten. Die weiteren Funktionen beziehen sich jeweils auf die Werte eines Attributes. Es sind dies: Mittelwert, Minimum und Maximum, Standardabweichung, Varianz, Häufigkeitsverteilung /LEVG 74/.

c) Konvertierungsfunktionen

Intern verschlüsselte oder abgekürzte Daten können in eine leicht lesbare Form umgewandelt werden. Daneben gibt es eine automatische Angleichung der Einheiten, d.h. der Benutzer wählt zu einem Attribut die Einheit für den Bericht (z.B. kg) und braucht sich dann nicht mehr zu kümmern, in welcher Einheit die Daten in der Datenbasis gespeichert sind /SKRH 77/.

2.5 Ausgabesteuerung

Bei der Berichtsdefinition (vgl. Abschnitt 2.1.2 b)) beschreibt der Anwender für den Bericht die logische Report-Seite (d.h. logische Seitenlänge, -breite), noch ohne an ein bestimmtes Ausgabegerät für den Bericht zu denken (Geräteunabhängigkeit der Berichtsdefinition). Damit ein Bericht auf verschiedenartigen Geräten ausgegeben werden kann, ist es notwendig, zwischen "(logischer) Report-Seite" und der "physikalischen Seite" des Berichts auf dem Ausgabemedium zu unterscheiden.



Zur Definition der physikalischen Seite sind die Charakteristika des jeweiligen Ausgabegeräts zu betrachten, d.h. sie ist gerätespezifisch: Zur Definition der physikalischen Seite eines Druckers, z.B. liegt es nahe, die Größe der Seiten des Endlosformulars zu verwenden; bei Sichtgeräten ist es meist sinnvoll die Bildschirmgröße zugrunde zu legen.

Aufgabe der Ausgabesteuerung ist nun, die Zuordnung Report-Seite - physikalische Seite (des jeweiligen Geräts) durchzuführen.

Die meisten Report-Generatoren erlauben es dem Anwender nicht, diese Zuordnung explizit zu beschreiben; i.a. ist eine logische Seite genau einer physikalischen Seite zugeordnet, wobei implizit angenommen wird, daß eine logische Seite in eine physikalische Seite paßt.

In gewissem Umfang erlaubt der Report-Generator vom SYSTEM 2000 eine Spezifizierung der Zuordnung von logischen und physikalischen Seiten. Es wird zwar noch immer vorausgesetzt, daß die logische Seite kleiner ist als die physikalische, aber es können mehrere logische Seiten auf eine physikalische Seite abgebildet werden; dies wird in Bild 5 erläutert.

Ein weitergehender Ansatz ist in /DANC 72/ zu finden: Es wird vorgeschlagen, mehrere Zuordnungen von Report-Seiten zu physikalischen Seiten vorzusehen, um so den spezifischen Eigenschaften verschiedener Ausgabegeräte - es wird vor allem zwischen "hard copy devices" und "soft copy devices" unterschieden - gerecht zu werden; dabei geht man soweit, daß Teile einer logischen Report-Seite verschiedenen physikalischen Seiten zugeordnet werden können. Dies wird an 2 Beispielen erläutert:

Die Zuordnung von Bild 6 a eignet sich für Ausgabe auf Papier (d.h. hard copy devices), da man hier die physikalischen Seiten wieder zusammenfügen kann; für Sichtgeräte dürfte sie ungünstig sein, da eine logische Berichtszeile nicht in einer physikalischen Seite ausgegeben wird. Entsprechend wird die Zuordnung von Bild 6 b besonders den Verhältnissen bei Sichtgeräten gerecht, da der Inhalt der logischen Seite ganz (wenn auch anders angeordnet) in einer physikalischen Seite untergebracht ist /DANC 72/.

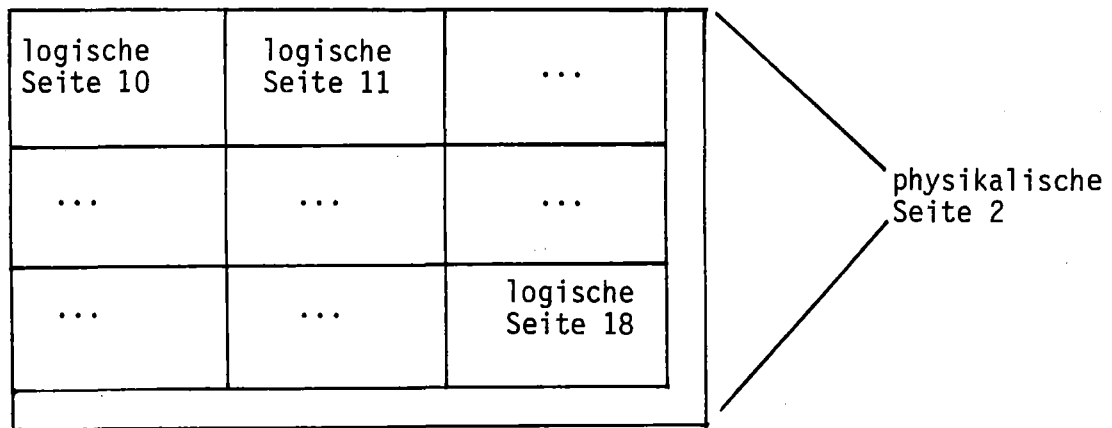
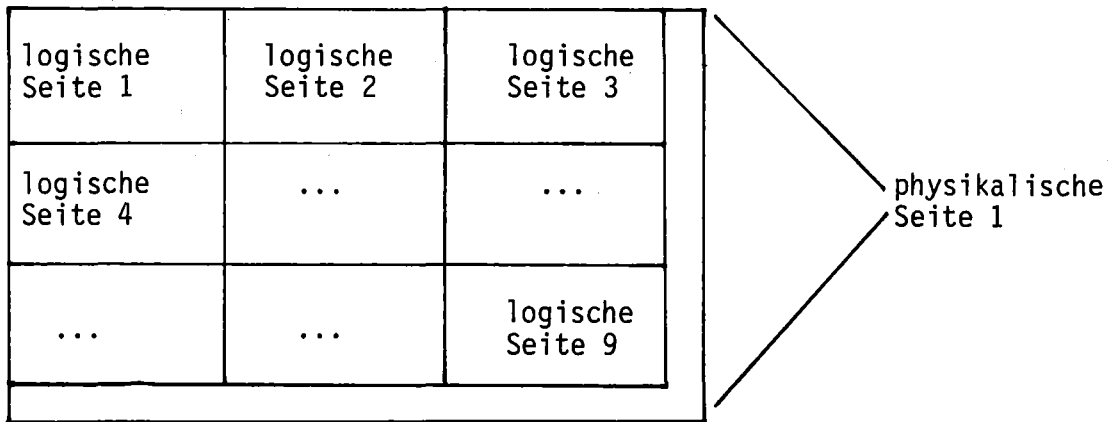
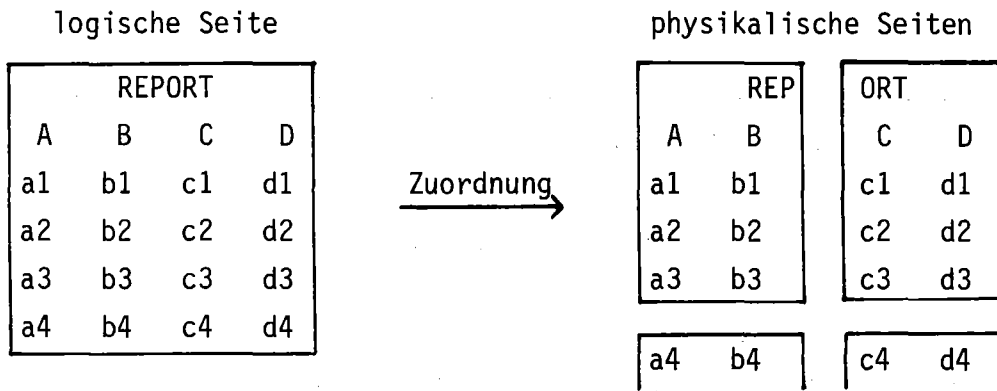
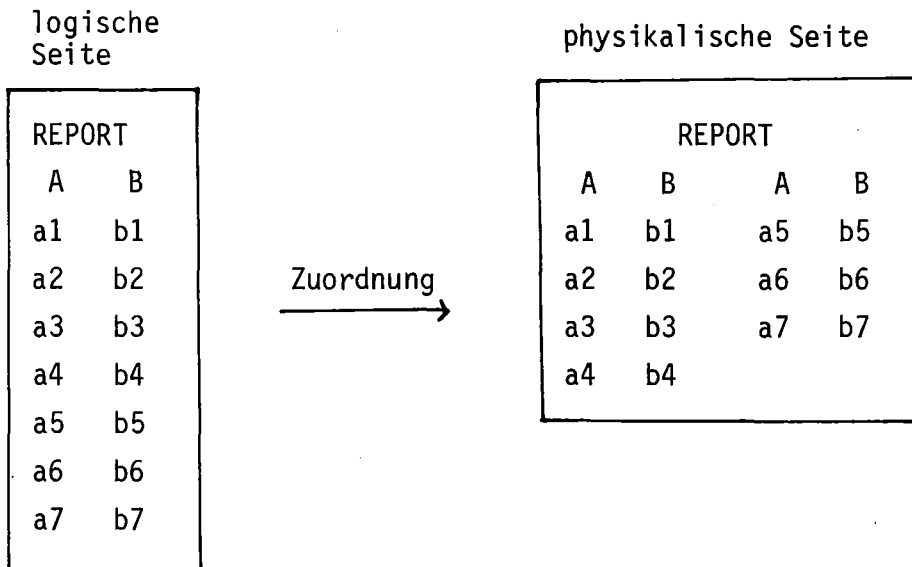


Bild 5: Beispiel einer Zuordnung von logischen Seiten (Seitengröße: 30x30) zu physikalischen Seiten (Seitengröße: 100x100)(aus /SYST 74/).



a) Logische Seite ist größer als physikalische Seite



b) Logische Seite ist kleiner als physikalische Seite

Bild 6: Zuordnungen von logischen Seiten zu physikalischen Seiten (aus /DANC 72/).

## 2.6 Zugriffsschutz

Mit dem Zugriff auf ein Report-Programm ist zugleich der Zugriff zu Daten einer Datenbasis verbunden. Zum Schutz vor unberechtigtem Zugriff zu Daten sind also Maßnahmen erforderlich, die gewährleisten, daß nur entsprechend autorisierte Benutzer Zugriff zu Report-Programmen haben (Bedienungsschutz).

Es besteht daneben auch die Forderung nach Datenschutz; dies ist ein allgemeines Datenbankproblem und wird daher hier nicht näher behandelt.

Darüberhinaus sind zusätzliche Maßnahmen erforderlich, um Informationen in einem Bericht zu schützen, die erst durch die Auswertung der Daten der Datenbasis durch das Report-Programm entstanden sind (und damit nicht Bestandteil der Datenbasis sind).

Nach den vorliegenden Unterlagen wird nur beim IMS Report-Generator versucht, entsprechende Möglichkeiten anzubieten. Bei Report-Definition kann hier spezifiziert werden, ob ein Benutzer seine Berechtigung

- zum Löschen der Report-Definition
- zur Modifikation der Report-Definition
- zum Auflisten der Report-Definition

durch Angabe eines Kennwortes nachzuweisen hat. Das Kennwort wird gegebenenfalls bei der Report-Definition festgelegt. Der Schutz vor unberechtigtem Ausführen oder Definieren eines Report-Programms ist im Rahmen des IMS-Gesamtsystems geregelt.

### 3. Schnittstellen

Ein Report-Generator ist im allgemeinen nicht als vollkommen selbständiges System denkbar, sondern er stellt eine Komponente innerhalb eines Informationssystem dar und übernimmt die Ausgabe von Daten an den Benutzer. Die Einbettung des Report-Generators in das Gesamtsystem und seine Schnittstellen zu anderen Komponenten verdeutlicht Bild 7:

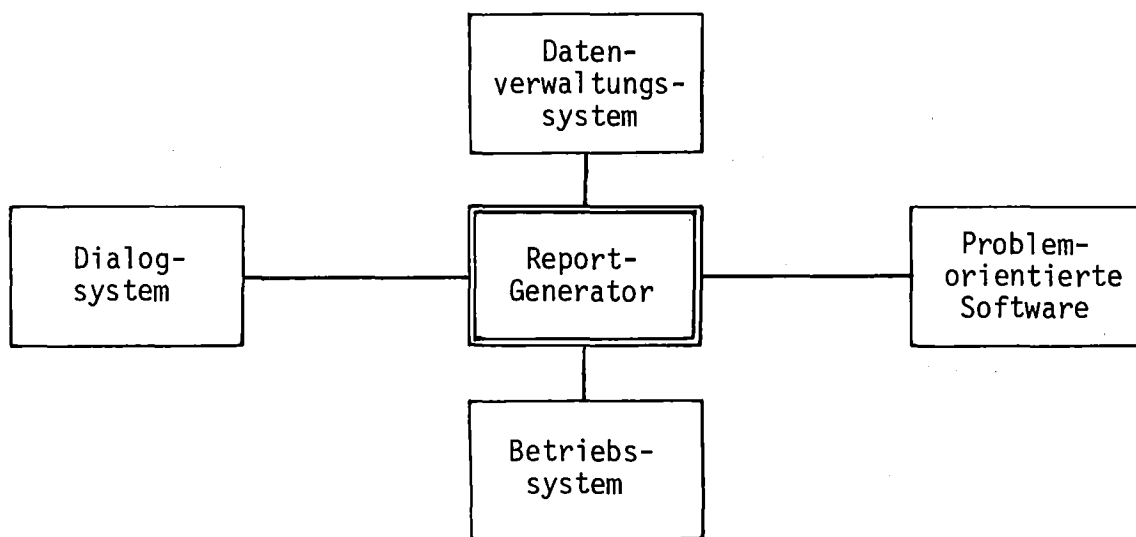


Bild 7: Schnittstellen eines Report-Generators

Es sind Schnittstellen zu folgenden Komponenten vorhanden:

- Dialogsystem  
Dieses wickelt den Dialog mit dem Benutzer ab.
- Datenverwaltungssystem  
stellt die angeforderten Daten zur Report-Generierung bereit.
- Problemorientierte Software  
Dem Anwender wird dadurch eine Möglichkeit zur Erweiterung des Report-Generators um eigene Funktionen gegeben.
- Betriebssystem.

Die einzelnen Schnittstellen werden anschließend funktional beschrieben.

### 3.1 Schnittstelle zum Dialogsystem

Der Benutzer gibt über Dialoge Kommandos zur Steuerung des Report-Generators ein.

Die Kommandos unterscheidet man nach der jeweiligen Funktion:

- Report-Definition und -Modifikation (vgl. Abschnitt 2.1, 2.2)  
Damit werden struktureller Aufbau und äußere Form des gewünschten Berichts festgelegt.
- Datenauswahl (vgl. Abschnitt 2.3)  
Die Daten, die der Bericht enthalten soll, werden spezifiziert. Dieser Teil der Dialog-Schnittstelle stellt die bei Datenverwaltungssystemen übliche Anfragesprache dar.
- Datenverknüpfung (vgl. Abschnitt 2.4)  
Anhand dieser Funktionen können die ausgewählten Daten vor der Ausgabe verknüpft werden.
- Ausgabesteuerung (vgl. Abschnitt 2.5)  
Festlegung der physikalischen Seitengröße.

Bei den meisten Report-Generatoren werden alle Kommandos und Parameter zu Beginn des Generator-Laufes eingegeben. Einzelne Systeme erlauben es, die jeweiligen Funktionen interaktiv zu bedienen.

Zur Dialogführung werden verschiedene Techniken angeboten. Diese kann man nach dem Grad der Unterstützung einteilen, die sie dem Benutzer bieten.

Wenig Unterstützung im Aufbau einer Dialogzeile und im Dialogablauf bietet die

- Statementweise Eingabe-Technik  
Der Benutzer bildet aus Schlüsselwörtern und Parametern nach einer festen Syntax einzelne Statements und baut daraus ein Report-Programm auf. Mit der Anweisungsfolge legt er auch den Dialogablauf fest (z.B.: QUERY/HP, FOIN/DIETZ).

Der Benutzer muß sich also in der Syntax der Anweisungen genau auskennen und muß wissen, aus welchen Teilen ein Report-Programm insgesamt aufgebaut ist. Weniger Detailkenntnisse verlangen folgende Techniken und bieten mehr Unterstützung bei der Dialogführung:

- Frage-Antwort-Technik

Die Parameter für die einzelnen Funktionen werden durch das System abgefragt. Die Reihenfolge der Abfrage ist vorgegeben. Der Benutzer hat bei jeder Frage die Möglichkeit, entweder seine Parameter einzugeben oder die Frage unbeantwortet zu lassen, wobei dann vom System Default-Werte eingesetzt werden (z.B.: IMS, PARLIS/mbp).

Werden die Fragen formularähnlich angeordnet, so daß der Benutzer im Dialog nur noch die freien Stellen des Formulars auszufüllen hat, dann nennt man diese Technik auch "fill-in-the-form" /KINP 73/ oder "fill-in-the-blanks" /MCLD 77/.

- Menue-Technik

Das System gibt dem Benutzer zunächst ein Menue der möglichen Dialogabschnitte (Funktionen) vor. Nachdem der Benutzer den gewünschten Dialogabschnitt gewählt hat, werden ihm die Kommandos und Parameter dazu angeboten. Der Benutzer stellt sich daraus sein Report-Programm zusammen. Der Benutzer kann sich jederzeit, so weit vom Dialogablauf sinnvoll, das Menue der möglichen Dialogabschnitte geben lassen und einen neuen Abschnitt auswählen (z.B.: IREG/GMD).

Durch Vorgabe der Fragen bzw. eines Menues wird vom System der Dialogablauf bestimmt bzw. eingegrenzt. Der Benutzer bekommt alle nötige Information zum Aufbau eines Report-Programmes per Dialog.

Die Anforderungen an die Dialogschnittstelle wachsen mit dem zu erwartenden Anstieg der gelegentlichen Benutzer ("casual user"), die keine oder sehr wenig Erfahrung im Umgang mit Computern aufweisen /CODE 74/. Angestrebt wird, diesen Benutzern einen Dialog in einer Sprache anzubieten, die der Umgangssprache nahe kommt /JARV 78/.

Als zusätzliche Hilfe ist es möglich, während eines Dialogs zu einem "Auskunft"-Modus ("HELP") zu verzweigen, der den Benutzer informiert über:

- Systemstatus
- weitere Kommandos, die zum gegebenen Zeitpunkt möglich sind
- Aufbau und Funktion von Kommandos.

Nachdem sich der Benutzer Klarheit verschafft hat, kann der Dialog an der unterbrochenen Stelle fortgesetzt werden /IREG 77, QUER 76, LEVG 74, KINP 73/. Bei IREG und in /KINP 73/ kann sich der Benutzer jederzeit über das aktuell definierte Berichtsbild informieren und so die Wirkung von Anweisungen verfolgen. Eine umgehende Korrektur des Report-Programms ist dadurch möglich.

### 3.2 Schnittstelle zum Datenverwaltungssystem

Zur Durchführung der in 2.1, 2.2 und 2.3 genannten Report-Generatorfunktionen sind u.a. folgende Operationen bzw. Informationen erforderlich.

- (1) Lesen von Daten aus der Datenbasis;  
Selektion der Report-Records gemäß Auswahlbedingung (Qualifikation) des Report-Programms
- (2) Extraktion der für den Bericht erforderlichen Attributwerte
- (3) Ordnen der Report-Records gemäß Sortieranweisungen
- (4) Aufbereitung, z.B. Formatierung der Daten für die Ausgabe.

Besonders für die Punkte (1), (2), (4) ist im allgemeinen der Zugriff



auf ein Daten-Dictionary erforderlich, das eine Beschreibung der Datenbasis enthält; hierzu gehören z.B.

Angaben zur Dateistruktur, Satzstruktur, Attributnamen, -typen, internen Darstellung.

Fähigkeiten zur Verwaltung von Dateien werden für Punkt (3) benötigt:

Unabhängig von einem speziellen Verfahren sind i.a. zum Ordnen der Report-Records beim Sortiervorgang größere Datenmengen als Zwischenergebnisse temporär auf Peripheriespeichern abzulegen. (Diese Zwischenergebnisse können Indexlisten sein oder auch die Report-Records nach einer von eventuell mehreren Sortierphasen).

Die Fähigkeit zum Zwischenspeichern muß auch dann gegeben sein, wenn aus einer Menge von Report-Records mehrere Berichte erzeugt werden sollen.

Es sind daher im allgemeinen für die Report-Erstellung Dateien einzurichten, zu verwalten und zu löschen (siehe Bild 8).

Report-Generatoren wie RPG /LESH 67/, GIRLS /POSJ 62/ und MARK IV /POSJ 68/ setzen kein Datenbanksystem voraus. Daher ist ein wesentlicher Bestandteil solcher Systeme ein Dateiverwaltungssystem und ein Daten-Dictionary. Das Dictionary kann wie bei GIRLS als "magnetic tape dictionary" gefordert sein; bei RPG dagegen ist die für den Bericht wesentliche Dictionary-Information im Report-Programm selbst enthalten. E/A-Operationen werden hier sequentiell durchgeführt, da diese Systeme noch Magnetbänder als Sekundärspeicher benutzen.

Da moderne Datenbanksysteme ein Dateiverwaltungssystem und ein Dictionary enthalten, werden neuere Report-Generatoren wie SYSTEM 2000, IMS, IREG, QUERY als integraler Bestandteil oder zumindest als utility betrachtet und beschrieben. Für diese Systeme liegen keine Angaben über eine Schnittstelle Report-Generator - Datenbanksystem vor. Man kann jedoch davon ausgehen, daß ein Report-Generator sich der (gewöhnlichen) Benutzerschnittstelle bedient. Die Operationen (1) und (2) werden dann vom Datenbanksystem (DBS) durchgeführt und sind nicht mehr notwendigerweise Bestandteil des Report-Generators.

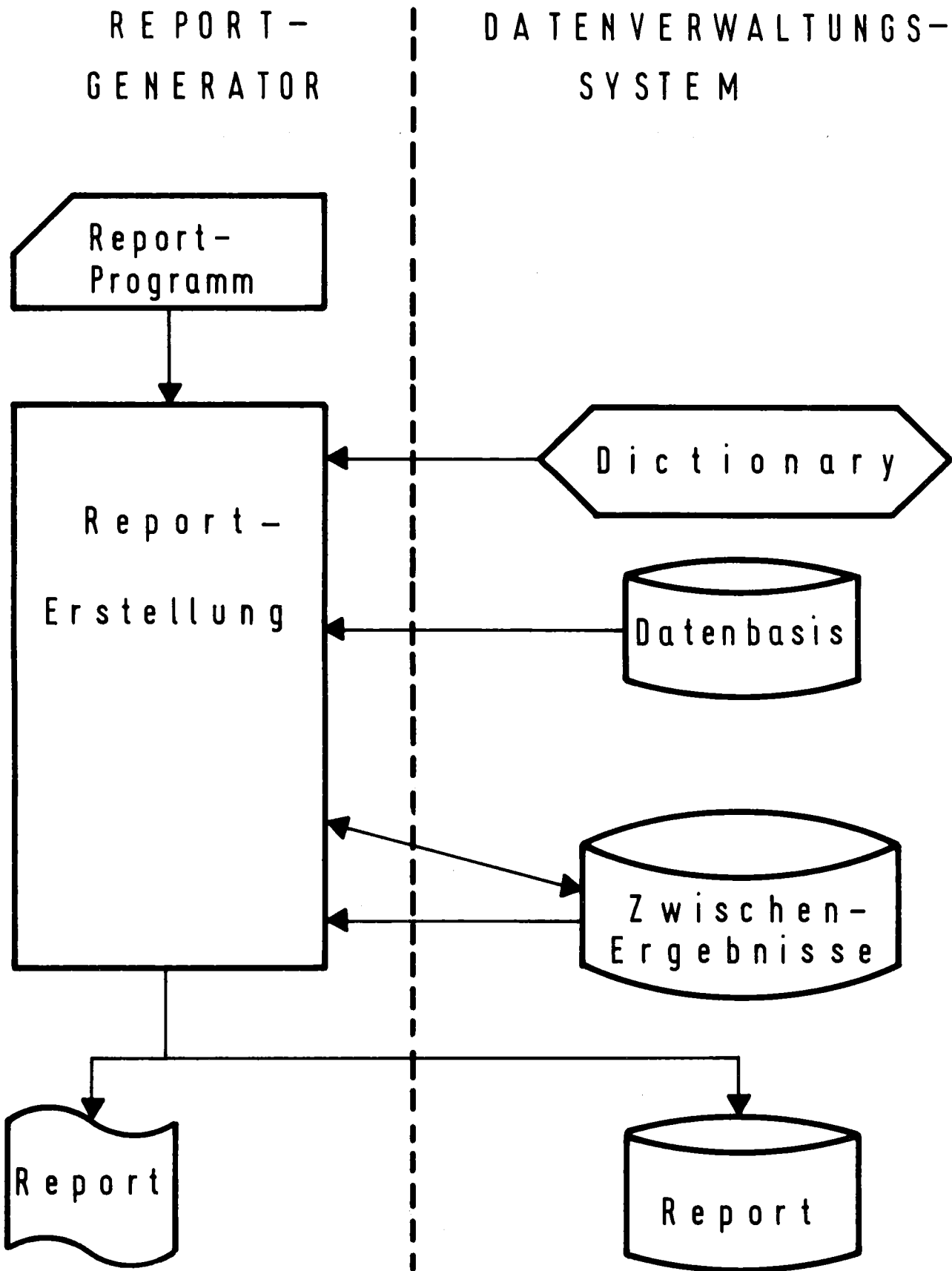


Bild 8: Zusammenspiel: Report-Generator / Datenverwaltungssystem

Sofern nicht bereits in der Benutzerschnittstelle vorhanden, sind noch zusätzlich für (4) Aufrufe zum Lesen aus dem Datenbanksystem-Dictionary erforderlich.

Unter Umständen kann das Sortieren von Report-Records (3) ebenfalls durch das Datenbanksystem übernommen werden; Systeme, die invertierte Listen z.B. als B-Bäume implementieren, sind auf Grund von B-Baum-Eigenschaften (vgl. /BAYR 72/) bereits mit dem Errichten von invertierten Listen in der Lage, die Report-Records nach einem bestimmten Kriterium geordnet ausgeben. In solchen Fällen könnte also dem Report-Generator ein Datenbanksystem-Aufruf "Sortiere Datei nach dem Attribut A" zur Verfügung stehen, der im wesentlichen in das DBS-Kommando "Errichte invertierte Liste" übersetzt wird, so daß auch die Sortierfunktion in das DBS verlagert wäre.

### 3.3 Schnittstelle zur problemorientierten Software

Der Benutzer hat die Möglichkeit, eigene Anwendungsprogramme durch den Report-Generator anzusprechen. Beim Aufruf müssen bestimmte Konventionen eingehalten werden. Der Funktionsumfang des Report-Generators kann durch die problemorientierte Software erweitert werden, was besonders zur Verknüpfung von Daten, d.h. der anwendungsspezifischen Auswertung sinnvoll erscheint.

Diese Schnittstelle wird von den meisten Report-Generatoren vernachlässigt; IREG z.B. bietet nur für Assembler-Unterprogramme eine Anschlußmöglichkeit.

### 3.4 Schnittstelle zum Betriebssystem

Der Report-Generator als Bestandteil der Systemsoftware benützt die Standard-Aufrufe des Betriebssystems. Nähere Angaben über Umfang und Art der zugrunde gelegten Betriebssystemfunktionen lassen sich zu den betrachteten Systemen nicht ableiten.

#### 4. Ergebnisse und Ausblick

Zusammenfassend können als wichtigste Erkenntnisse aus der zurückliegenden Arbeit genannt werden:

- Die untersuchten Systeme bieten ausreichende Funktionen zur Report-Definition und Datenauswahl. Der Sprachkomfort läßt dabei meist zu wünschen übrig.
- Zur Datenverknüpfung stehen meist nur die vier Grundrechenarten zur Verfügung und eine Möglichkeit zum Anschluß von Anwenderprogrammen ist bis auf Ausnahmen nicht vorhanden.
- Bei fast allen Systemen fehlt die Möglichkeit einer Trennung zwischen logischer und physikalischer Seite. Eine Schnittstelle zur graphischen Ausgabe fehlt.
- Für Kleinrechner auf technisch-wissenschaftlichem Gebiet sind nur wenige Systeme auf dem Markt.

Die obigen Schlüsse wurden aufgrund globaler Gesichtspunkte gezogen.

Die Anforderungen an einen Report-Generator sollen im nächsten Schritt gesammelt und den hier genannten Funktionen und Schnittstellen zugeordnet werden.

## 5. Literatur

Literaturstellen, die einen unmittelbaren Bezug zum Thema "Report-Generator" haben, sind mit "\*" gekennzeichnet. Bei den ersten 7 Referenzen handelt es sich um Systembeschreibungen, die getrennt von den allgemeinen Literaturstellen aufgeführt sind.

- \*/FOIN/ FOIN/MAGICS, Formular Interpreter im MAGICS-System.  
DIETZ-621 MAGICS-Systembeschreibung, Nr. 2-7509-01-010,  
Mühlheim-Ruhr
- \*/CAGE 76/ Cagley, E.M., et al.: Information Management System Reference  
Manual (IMS). Microfiche PB-258915, Mathematics and Computation  
Lab. (EDM), Washington D.C. (Oct. 1976), pp. 103-121
- \*/IREG 77/ IREG - Interaktiver Report-Generator des Datenbanksystems  
FIDAS, Benutzeranleitung. Gesellschaft für Mathematik und Daten-  
verarbeitung mbH, Bonn (GMD), (Mai 1977)
- /FIDA 76/ FIDAS - Formularorientiertes Interaktives Datenbanksystem  
(Vers. 3), Einführung. Gesellschaft für Mathematik und Datenver-  
arbeitung mbH, Bonn (GMD), (Jan. 1976)
- \*/PARL/ PARLIS - System zur Erzeugung von Listen und Formularen. mbp -  
Mathematischer Beratungs- und Programmierungsdienst GmbH, Dort-  
mund
- \*/QUER 76/ QUERY - HP 3000 Series II Computer System, Reference Manual.  
Prod.No. 32216A, Hewlett Packard, Santa Clara, California (1976)
- \*/SYST 74/ SYSTEM 2000 - Report Writer Feature.  
MRI Systems Corporation, Austin/Texas, (1974)
- /AHRK 76/ Ahrens, Klaus-Dietrich: Programmieren ohne Programmierer? - Re-  
duktion des Testaufwands durch BAS. Online, 14 (April 1976),  
S. 248-251
- /BAYR 72/ Bayer, R., McCreight, E.: Organization and Maintenance of Large  
Ordered Indexes. Acta Informatica, 1 (1972) pp. 173-189
- /CODE 74/ Codd, E.F.: Recent Investigations in Relational Data Base  
Systems. In: Rosenfeld, Jack L. (Hrsg.): Information Processing  
74, Stockholm, Sweden, August 5-10, 1974.  
Amsterdam: North Holland Publishing Company 1974, pp. 1017-1021
- \*/DANC 72/ Dana, C., Presser, L.: An Information Structure for Data Base  
and Device independent Report Generation. In: AFIPS Conference  
Proceedings, Fall Joint Computer Conference, Anaheim, California,  
Dec. 5-7, 1972. Montvale, New Jersey: AFIPS Press, Vol. 41, 1972,  
pp. 1111-1116

- \* /FRIL 67/ Friedberg, L.M.: RPG: The Coming of Age. Datamation, 13 (1967) No. 6, pp. 29-31
- /FRYJ 76/ Fry, J.P., Sibley, E.H.: Evolution of Data-Base Management Systems. Computing Surveys, 8 (March 1976) No. 1, pp. 7-42
- \* /HIRL 76/ Hirschmann, L., Grishman, R., Sager, N.: From Text to structured Information - Automatic Processing of Medical Reports. In: AFIPS Conference Proceedings, National Computer Conference, New York City, June 7-10, 1976. Montvale, New Jersey: AFIPS Press, Vol. 45, 1976, pp. 267-275
- /JARV 78/ Jarsch, V., Landmark, K., Müller, W.: Grundlagen und Entwurfsmethodik eines Dialogsystems. KfK-Bericht 2256, Jan. 1978
- /JOYJ 76/ Joyce, J.D., Oliver, N.N: REGIS - A Relational Information System with Graphics and Statistics. In: AFIPS Conference Proceedings, National Computer Conference, New York City, June 7-10, 1976. Montvale, New Jersey: AFIPS Press, Vol. 45, 1976, pp. 839-844
- \* /KINP 73/ King, P.F.; Shemer, J.E.: ARS - An Interactive Reporting System. In: Proceedings of ACM SIGPLAN-SIGIR Interface Meeting, Gaithersburg, Maryland, Nov. 4-6, 1973. SIGPLAN Notices, 10 (1975) No. 1, pp. 161-166
- \* /LESH 67/ Leslie, H.: The Report Program Generator. Datamation, 13 (1967), No. 6, pp. 26-28
- \* /LEVG 74/ Levitt, G., Stewart, D.H., Yormark, B.: A Prototype System for Interactive Data Analysis. In: AFIPS Conference Proceedings, National Computer Conference, Chicago, Illinois, May 6-10, 1974. Montvale, New Jersey: AFIPS Press, Vol 43, 1974, pp. 63-69
- \* /LONL 62/ Longo, L.F.: SURGE: A Recoding of the COBOL Merchandise Control Algorithm. Communications of the ACM, 5 (1962), No. 2, pp. 98-100
- \* /LUCH 74/ Lucas, H.C.: A CRT Report Generating System. Communications of the ACM, 17 (1974) No. 1, pp. 47-48
- /MCGW 59/ McGee, W.C.: Generalization: Key to Successful Electronic Data Processing. Journal of the ACM, 6 (Jan. 1959) No. 1, pp. 1-23
- /MCLD 75/ McLeod, D., Meldman, M.: RISS - A Generalized Minicomputer Relational Data Base Management System. In: AFIPS Conference Proceedings, National Computer Conference, Anaheim, California, May 19-22, 1975. Montvale, New Jersey: AFIPS Press, Vol. 44, 1975, pp. 397-402
- \* /MCLD 77/ McLeod, D., Meldman, M.: SELECT: An Interactive Minicomputer Subsystem für the Generation of Spezialized Narrative Text. Computers and Biomedical Research, 10 (Apr. 1977), pp. 91-99

- /MYLJ 75/ Mylopoulos, J., Schuster, S., Tschritzis, D.: A Multi-Level Relational System. In: AFIPS Conference Proceedings, National Computer Conference, Anaheim, California, May 19-22, 1975. Montvale, New Jersey: AFIPS Press, Vol. 44, 1975, pp. 403-408
- \*/POSJ 62/ Postley, J.A., Buetell, T.D.: Generalized Information Retrieval and Listing System. Datamation, 8 (Dec 1962), No. 12, pp. 22-25
- \*/POSJ 68/ Postley, J.A.: The MARK IV System. Datamation, 14 (Jan. 1968) No. 1, pp. 28-30
- \*/SKRH 77/ Skronn, H.-J.: ESEL - ein Listengenerator für eine Datenbank. Angewandte Informatik, 19 (1977) Nr. 3, S. 109-114
- /WEDH 74/ Wedekind, H.: Datenbanksysteme I. 1. ed. Zürich: Bibliographisches Institut AG, 1974