DOI: 10.20378/irb-57793

BAMBERGER

BEITRÄGE ZUR WIRTSCHAFTSINFORMATIK

ISSN 0937-3349

Nr. 3/1990

# Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

Walter Augsburger
Helge Rieder
Jürgen Schwab

Erscheint als Tagungsbeitrag in:

ISI '90, 1. Internationales Symposium für Informationswissenschaft
17.-19. Oktober 1990, Konstanz
"Pragmatische Aspekte beim Entwurf und Betrieb von Informationssystemen"

Alle Rechte vorbehalten

(c) 1990 by O. Forst, B. Sinz

OTTO-FRIEDRICH-UNIVERSITÄT BAMBERG

# Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen

**Abstract**

Zeitabhängige, numerische Daten über das Unternehmen und aus dem Unternehmensumfeld werden zur Unternehmensplanung und zum Controlling oft nur unzureichend genutzt. Zu den Gründen hierfür zählen eine nicht an den Wünschen des Endbenutzers orientierte Auswertung und Aufbereitung. Der vorliegende Tagungsbeitrag stellt das im Projekt EISREVU entwickelte, auf der Struktur über und von Zeitreihenobjekten aufbauende und strukturerzeugende, Modellierungssystem vor. Es ermöglicht dem Endanwender einen individuellen, verteilten und netzwerktransparenten Aufbau von unterschiedlichen Kennzahlensystemen einschl. der Definition und Anwendung von formalen Bewertungskriterien für Kennzahlen. Daneben wird ein allgemeiner Überblick über Aufbau und Anwendung von Zeitreihenobjekten und die Sprache zur Definition von komplexen Zeitreihenobjekten gegeben.

## 1. Ausgangspunkt

Die flexible Definition, Integration und Verwaltung ausschließlich numerischer Daten in betrieblichen Informationssystemen führt in der wissenschaftlichen Diskussion eher ein Schattendasein; ebenso deren Auswertung und sprachliche Interpretation. Dabei stellen numerische Daten wichtige, jedoch interpretationsbedürftige Informationsträger für Entscheidungen der Unternehmensleitung, der zweiten Führungsebene und für das Controlling dar. Zu dieser Fragestellung wird von unserem Lehrstuhl in Zusammenarbeit mit zwei Unternehmen aus der Energieversorgung das im folgenden teilweise vorgestellte Projekt EISREVU durchgeführt.

Der vorliegende Beitrag stellt das Konzept der "Zeitreihenobjekte", ihrer Strukturierung und darauf aufbauend das System zur Gewinnung von Informationen aus numerischen, zeitabhängigen Daten vor. Er ist die Beschreibung eines auf Grundlage eines theoretischen Konzepts implementierten Prototyps. Aufgabe des Systems ist es, periodisch erhobene numerische Daten aus dem Unternehmen und dem Umternehmensumfeld zu bewertbaren Kennzahlen umzuformen und diese automatisch für den Entscheidungsträger vorzubewerten. Dabei kann das System auch von sich aus aktiv werden. Der Unterschied zu vergleichbaren Systemen (z.B. teilautomatischer Bilanzbewertung) [Schä 88] besteht darin, daß der Anwender kein fertig

---

Das Projekt EISREVU wird finanziell und durch fachliche Zusammenarbeit von den Unternehmen Mainkraftwerke AG Frankfurt-Höchst (MKW) und Energieversorgung Oberfranken AG (EVO) gefördert.

strukturiertes, sondern eine flexibles, nach eigenen Bedürfnissen einsetzbares Werkzeug zur eigenen Modellbildung erhält. "

## 2. Adressaten und Zielvorstellungen des Projekts

Anwender-Zielgruppen des Systems sind Entscheidungsträger aus der zweiten Führungsebene und dem Top-Management (zur Differenzierung Modellierer/Anwender siehe [Aug 89]). Durch Aggregation, Selektion, Bewertung numerischer Daten und deren Überführung in leicht verständliche Texte ist es möglich, Informationen, die in den "Datengräbern" einer Unternehmung zwar prinzipiell vorhanden, aber aufgrund mangelnder Aufbereitung und/oder Verdichtung nicht zu verstehen sind, für Entscheidungsträger bereitzustellen. Auch praxisorientierten Werken [Oeh 78] sind kaum Richtwerte für eine optimale Größe selbst der wichtigsten Kennzahlen zu entnehmen. Ein Informationssystem auf der Basis von Kennzahlen hat sich deshalb in erster Linie mit deren zeitlicher Entwicklung zu befassen, um Entwicklungstendenzen und Auffälligkeiten im Unternehmen und in der Gesamtwirtschaft erkennen zu können. Um auch Prognosen aufgrund neuer Entwicklungen schnell - und nach Möglichkeit automatisch - revidieren zu können, ist die Etablierung eines quasi permanenten Zeitreihenverwaltungssystems (im folgenden ZRV genannt) erforderlich.

## 3. Konzeption des Zeitreihenverwaltungssystems (ZRV)

Die EISREVU-Systemkonzeption folgt der Entwicklung zur Trennung von Benutzersicht und Datenstruktur. Anders als klassische Datenbanksysteme, die nur eine logische und keine systemtechnische Trennung beinhalten, anders als Front-Ends (wie z.B. [Fre 84]), die ihre Clients auf primitiven Endbenutzerschnittstellen aufsetzen, hat EISREVU eine Client-Server-Architektur mit einer ausschließlichen Maschine-Maschine-Kopplung (s. hierzu auch [Kuh 89], [Her 86] u.a.) und einer gewissen lokalen Intelligenz im Client. Aus diesem Grund enthält das ZRV keine direkte Endbenutzer-Schnittstelle, sondern wird von anderen Programmen über Prozeßkommunikation angesprochen. Der Endbenutzer kommuniziert dabei direkt mit einem anderen Prozeß, normalerweise auf seinem lokalen Rechner, der ihm eine graphische Oberfläche bietet und der seinerseits wieder mit den ZRV-Servern kommuniziert. Zusätzlich sind weitere Anwendungssyteme als Clients denkbar.

Abb. 1 zeigt netzwerktransparente ZRV-Server. In einem Netzwerk können theoretisch beliebig viele Serverprozesse gleichzeitig arbeiten. Ein ZRV-Server kann mit mehreren und ggf. auch verschiedenartigen Clients kommunizieren. Werte von Zeitreihen innerhalb des ZRV können als Ausgangsinformation für ein Prognosesystem und/oder entscheidungsunterstützendes System auf Basis eines Expertensystems oder eines künstlichen neuronalen Netzes dienen. Des weiteren ist ein Client zur automatischen Generierung natürlichsprachlicher Berichte denkbar. So könnte beispielsweise zu jeder "Ausnahmesituation" eine Meldung von einer Überwachungsfunktion (Demon) generiert werden: zeigt ein oder zeigen mehrere Demons auf Werte, die außerhalb eines "normalen", vom Endbenutzer definierbaren Wertebereichs liegen, so wird für das entsprechende Zeitreihenobjekt ein Bericht generiert. Dieser setzt sich aus folgenden Informationsquellen zusammen:

* der dem Zeitreihenobjekt assoziierten Beschreibung.
* Textstücken, die an die Bewertungsskalen der Demons geheftet sind und bei entsprechenden Demonergebnissen aktiviert werden.

Der Client zur Berichtgenerierung sollte dabei wählweise lineraen, druckfähigen Text oder Hypertext erzeugen.

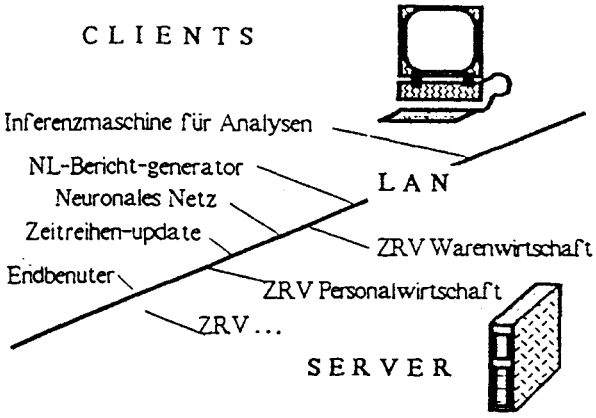



Abb. 1: ZRV-Server und ihre Clients

## 4. EISREVU-Zeitreihenobjekte und Modelle

### 4. 1. Zeitreihenobjekte und Kennzahlen

Inhaltliche Aufgabe des Konzepts der Zeitreihenobjekte ist die Verdichtung einer Vielzahl von Basisdaten auf eine überschaubare Anzahl von Maßzahlen. Dies ist in der Betriebswirtschaftslehre klassischerweise die Aufgabe der Kennzahlensysteme (s. z.B. [Mey 76]). Der Klassiker unter derartigen Systemen, speziell für die Spitzenkennzahlen, ist dabei das "Du Pont System of Financial Control" [Ame 50]. Daneben existieren in der Literatur Vorschläge zu Kennzahlensystemen speziell für Unternehmensteilbereiche, wie z.B. für die Materialwirtschaft [BIF 80, Blä 78], die Datenverarbeitung [Non 89], branchenspezifische Ansätze [Kup 82] u.ä.. Wie in [ZAI 88] dargestellt, lassen sich allgemeine Kennzahlensysteme für die interne Kennzahlenanalyse normalerweise nicht direkt auf die individuelle Struktur eines Unternehmens anpassen. Dazu kommen individuelle Benutzerpräferenzen: so wird ein Endanwender nicht zufrieden sein, sollten "seine" Kennzahlen nicht mit dem vorgegebenen System modellierbar sein.

### 4. 2. Konzept des Modellierungssystems

Die Konsequenz aus diesen Überlegungen ist die Konzeption und Realisierung eines Modellierungssystems, mit dem der modellierende Endbenutzer selbst sein individuelles Modell erstellen kann. Bei der

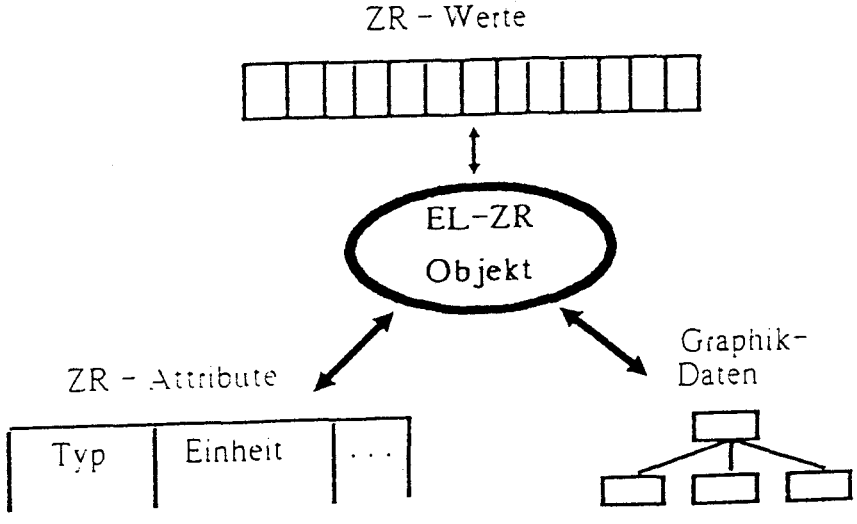



Abb. 2: Aufbau eines elementaren Zeitreihenobjekts

Konstruktion eines konkreten Modells wird dabei jeder von außen in das System eingeführten betriebswirtschaftlichen Größe ein elementares Zeitreihenobjekt zugeordnet. Elementare Zeitreihenobjekte besitzen, wie in Abb. 2 dargestellt, eine innere Struktur mit identifizierenden Namen und einer Zeitreihe mit Zusatzangaben (Einheit, Periodizität, etc. ).

Elementare Zeitreihenobjekte erhalten ihre Daten entweder automatisch über einen "Update-Client" oder durch manuelle Eingabe über den Client zur Benutzerinteraktion. Quellen für den "Update Client" sind die operativen Systeme des Unternehmens, amtliche Statistiken [Stau 85] auf Datenträgern sowie Daten aus Betriebsvergleichen [Reh 88], etc..

### 4. 3. Die Komponenten eines höheren Zeitreihenobjekts

Durch die **Bildungsregel** (s. Abb. 3) ist festgelegt, wie die ZR-Werte errechnet werden. Zu diesem Zweck wurde eine Beschreibungssprache entwikkelt, die modellierungstechnisch auf der Ebene der Regelsprachen von Expertensystemen angesiedelt und in Kap. 7. im Detail beschrieben ist. Die Bildungsregel eines jeden Zeitreihenobjekts wird in dieser Beschreibungssprache formuliert.

**ZR-Attribute** beinhalten Zusatzinformationen, wie Periodizität, Typ etc.

Die **Graphik-Komponente** speichert Informationen zur Anordnung der Zeitreihenobjekte auf dem Bildschirm. Diese Informationen werden vom Benutzer-Interaktions-Client ausgewertet. Die Darstellung der Zeitreihenobjekte erfolgt in einer planaren Graphik.

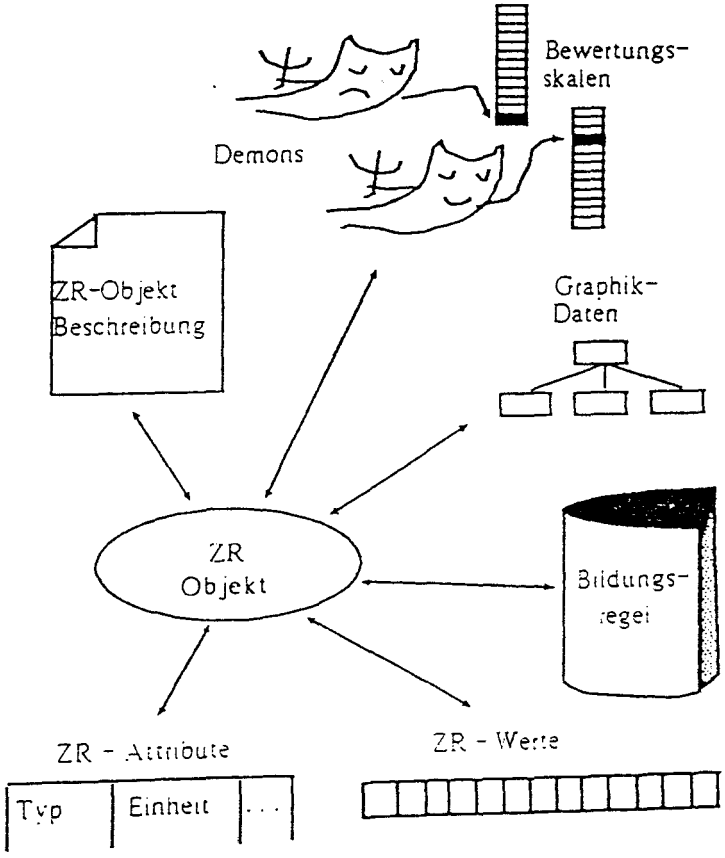



Abb. 3: Aufbau eines höheren ZR-Objekts

Die **ZR-Objekt Beschreibung** ist eine Art Notizzettel für Anmerkungen zum jeweiligen ZR-Objekt.

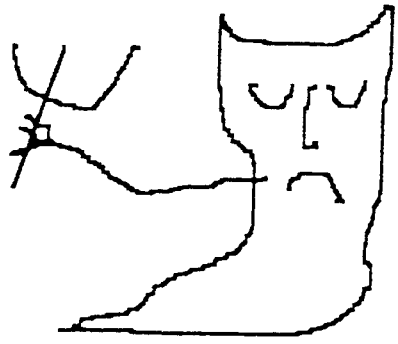
Als Werkzeug für die (Vor-)Bewertung und Überwachung von numerischen Größen in Zeitreihenform stehen die **Demons** zur Verfügung. Bei der Beurteilung betriebswirtschaftlicher Größen tritt das Problem auf, daß in der einschlägigen Literatur Bewertungskriterien oft nur sehr vage formuliert sind. Häufig nehmen in Standardwerken zur Bilanzanalyse (vgl.

[Lef 77] u.a.) die Berechnungsalgorithmen für Kennzahlen einen vielfachen Seitenumfang gegenüber den Bewertungskriterien ein. Selbst "Kochbücher" wie [BIF 80] geben selten konkrete Soll-Werte für Kennzahlen an. Die einzige Möglichkeit, die zeitliche Entwicklung von Kennzahlen in Regeln zu fassen, ist somit die Betrachtung der zeitlichen Entwicklung (s. [Schä 88]). Ein anderer Gesichtspunkt des Demon-Konzepts ist, daß sich die in verschiedenen Unternehmen eingesetzten Kennzahlensysteme aufgrund individueller Gegebenheiten stark voneinander unterscheiden, und da jeder Analytiker normalerweise seine eigenen Kriterien zum Einsatz bringen will, ist es notwendig, dem Benutzer eine flexible Modellierung von Datenbewertungsfunktionen zu ermöglichen.

Ein EISREVU-Demon führt diese Bewertung durch und ist normalerweise einem höheren Zeitreihenobjekt zugeordnet. Er besteht aus einem Verfahren, das jeweils nach up-dates an dem ihm zugeordneten Zeitreihenobjekt aktiviert wird. Nach der Aktivierung prüft der Demon, ob die ZR-Werte Bedingungen für die Auslösung einer Meldung an den Benutzer erfüllen. Zur Analyse der zugeordneten Zeitreihen kann sich ein Demon einer Reihe von vorgegebenen Verfahren bedienen. Der Demon kann z.B. feststellen, ob der letzte Zeitreihenwert einen vom Benutzer definierten Grenzwert überschritten hat. Treffen die Bedingungen für die Auslösung einer Meldung zu, so gibt der Demon vom Benutzer vordefinierte Meldungen aus, die dieser wiederum nach verschiedenen Gesichtspunkten abfragen kann.

Abb. 4 zeigt einen Demon KST_UEBERW zur Überwachung der Kostenstellenkosten. Falls die Werte der Zeitreihe eines des höheren Zeitreihenobjekts KST_Lager im Vergleich zum letzten Wert um mehr als 5 Prozent gestiegen sind, gibt der Demon die Meldung "Kostenstellenkosten sind zu stark gestiegen" und die Werte der Zeitreihen aus.

```
DEMON_DESCR ( KST_UEBERW )
BIND_TO KST_Lager
IF Steigung ( KST_Lager > 0.05 )
THEN
    Meldung ("Kostenstellenkosten
              sind zu stark gestiegen")
AND
Liste ( [KST_Lager])
```



KST Lager: 1025 1150 1328 1419 1589 1730

Abb. 4: Überwachung der Lagerkosten (leicht gekürzte Syntax)

### 4. 4. Globale Modelle und Strukturmodelle

Der Benutzer faßt Zeitreihenobjekte nach sachlogischen Gesichtspunkten in globalen Modellen und Strukturmodellen zusammen. Ein globales Modell ist ein in sich abgeschlossenes System. Es beinhaltet elementare Zeitreihenobjekte und Strukturmodelle über höhere Zeitreihenobjekte. Globale Modelle sind voneinander unabhängig. Es können mehrere Strukturmodelle in einem globalen Modell und somit mehrere Strukturmodelle über denselben elementaren Zeitreihenobjekten existieren. Ein Strukturmodell besteht aus höheren Zeitreihenobjekten. Es hat die Struktur von übereinandergelagerten, transponierten Wurzelbäumen mit benannten Kanten.

## 5. Der ZRV-Server als abstrakte Maschine

Der ZRV-Server wird im folgenden als abstrakte Maschine, die Befehle entgegennimmt und Antworten generiert, beschrieben. Der folgende Überblick über die Funktionalität eines ZRV-Servers beschreibt die Interaktionen, die zwischen dem Server und möglichen Clients ausgeführt werden können.

### Verwaltung von Zeitreihenobjekten

Hierzu zählen Befehle zum Anlegen und Löschen von elementaren und höheren Zeitreihenobjekten. Beim Anlegen eines höheren Zeitreihenobjekts ist dessen Bildungsregel mitzugeben. Weiterhin zählen hierzu Befehle zum Anlegen, Anfordern und Löschen der natürlichsprachlichen Beschreibung und der Demons eines Zeitreihenobjekts.

### Zeitreihenobjekte mit Inhalt füllen

**Elementare Zeitreihenobjekte** werden gefüllt, indem die Werte eines definierten Zeitraums, ggf. auch zum Anhängen an einen vorhandenen Bestand dem ZRV zusammen mit dem jeweiligen Befehl übergeben werden. Beim Füllen von **höheren Zeitreihenobjekten** werden keine Daten übergeben, sondern es wird die Bildungsregel des jeweiligen Zeitreihenobjekts aktiviert. Auf Wunsch können alle Zeitreihenobjekte eines Strukturmodells oder auch eines globalen Modells (in sinnvoller Reihenfolge) neu berechnet werden.

### Inhalt von Zeitreihen ausgeben

Es existieren Befehle zur Ausgabe aller Werte eines Zeitreihenobjekts, des Werts zu einem Zeitpunkt, der Werte eines Zeitintervalls, der Anzahl der vorhandenen Werte sowie von den Werten der Attribute.

### Befehle zur Verwaltung von (globalen) Modellen

Diese Befehle ermöglichen ein neues globales Modell anzulegen und wieder zu löschen.

### Befehle zur Verwaltung der Strukturmodelle einschl. ihrer ZR-objekte

Diese Befehle ermöglichen es, ein neues Strukturmodell anzulegen und wieder zu löschen. Zusätzlich dazu existieren Befehle, die ein im Hintergrund vorhandenes Strukturmodell (mit seinen höheren Zeitreihenobjekten) in den aktuell bearbeitbaren Bereich laden bzw. aus dem bearbeitbaren Bereich entfernen (ohne dabei das Strukturmodell zu löschen). Dies macht z.B. dann Sinn, wenn zu einem globalen Modell mehr Strukturmodelle existieren als auf dem Bildschirm sinnvoll anzeigbar sind.

### Konfiguration von Zeitreihenobjekten in Modellen und Strukturmodellen

Die Befehle dieser Familie liefern Listen von elementaren und/oder höheren Zeitreihenobjekten in einem globalen Modell oder einem Strukturmodell und eine Liste von Strukturmodellen in einem globalen Modell.

## 6. Aufgaben der Beschreibungssprache für höhere Zeitreihenobjekte

Für den ZRV-Server ist ein höheres Zeitreihenobjekt durch die Bildungsregel, die als Argument des Befehls "höheres ZR-Objekt erzeugen" mitgegeben wird, definiert. Diese Bildugsregel in der Beschreibungssprache direkt zu formulieren, wird in der derzeitigen EISREVU-Implementierung innerhalb des Endbenutzer-Clients als Expertenmodus angeboten. Die Schnittstelle allein würde jedoch der Philosophie des EISREVU-Systems widersprechen, nach der der modellierende Benutzer selbst die Definition seiner Zeitreihenobjekte und die Komposition seiner Modelle vornehmen können muß. Wie in Abb. 5 dargestellt, wird die Beschreibungssprache im Endausbau des EISREVU-Systems von einem graphischen Editor (als Teil des Clients zur Benutzerinteraktion) erzeugt. Diese wird dann wiederum mit dem Beschreibungssprachencompiler in vom ZRV benutzbaren Prolog-Code übersetzt.

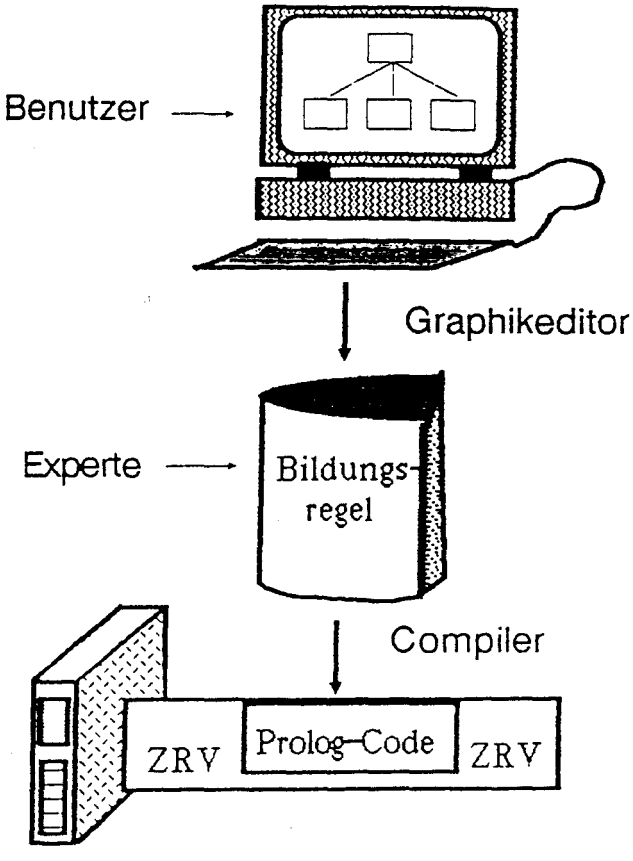



Abb. 5: Erstellung höherer Zeitreihenobjekte

Durch die interne Architektur des Servers werden im Prolog-Code des ZRV zur Laufzeit des Programms nicht nur Fakten sondern auch Regeln modifiziert. Durch die damit erfolgte defacto Aufhebung der Trennung von Programm und Daten in einem Prolog-System [Bra 87] können die Regeln zur Beschreibung höherer Zeitreihenobjekte (nach deren Übersetzung in Prolog-Syntax) sofort in das laufende Zeitreihenverwaltungssystem geladen, bzw. diese bei Laufzeit modifiziert werden. Hiermit unterscheidet sich der vorgestellte Ansatz von vielen wissensbasierten Systemen auf Prolog-Basis und allen Systemen auf Prolog-Dialekten wie Turbo-Prolog, die zur Laufzeit nur Fakten, aber keine Regeln modifizieren können.

## 7. Die Syntax der Beschreibungssprache

Aus Gründen des Gesamtumfangs dieses Beitrags ist die folgende Darstellung der Beschreibungssprache nicht ganz vollständig. Ebenso muß leider auf eine Darstellung der Syntax des Prolog-Compilats verzichtet werden.

### 7. 1. Grundlegende Struktur

```
höheres_Zeitreihenobjekt
     : object_def expression
```

Die Beschreibung eines höheren Zeitreihenobjekts besteht aus zwei Teilen: der erste Teil beinhaltet die Objektdefinition des neuen Zeitreihen-Objekts. Er entspricht in der Grammatikbeschreibung der object_def und beginnt mit dem Schlüsselwort "NODE_DESCR". Der zweite Teil beinhaltet die eigentliche Bildungsregel. Sie entspricht in der Grammatikbeschreibung einer expression und beinhaltet die Berechnungsvorschrift für die Werte des jeweiligen höheren Zeitreihenobjekts.

```
object_def
     : "NODE_DESCR" object model_types
```

Die Objektbeschreibung <object> besteht aus den terminalen Symbolen für die Definition der Parameter

```
objekt
     : ( name level type unit period )
```

| | |
|---|---|
| name: | Eindeutiger Identifikator des Objekts, |
| level: | Niveau des ZR-objekts - wird benötigt, um Zyklen bei der Definition von Zeitreihenobjekten zu vermeiden und entspricht einer Schicht in streng hierarchischen Systemen |
| type: | Wert, Verhältnis oder Index, |
| unit: | DM, $, etc., |
| period: | Tag, Monat, Vierteljahr, Halbjahr oder Jahr |

und der Definition der Zugehörigkeit zum globalen Modell und Strukturmodell in model_types

```
model_types
     : "GLOB_MODEL" STRING "STRUCT_MODEL" STRING
```

Beispiel:

```
NODE_DESCR (MA_Einkauf 27 value DM MONTH)
     GLOB_MODEL LAGER_BA
     STRUCT_MODEL Mitarbeiter ..
```

## 7. 2. Aufbau der Berechnungsvorschrift

Die Berechnungsvorschrift für die Werte eines höheren ZR-Objekts wird in der Grammatik der Beschreibungssprache <expression> genannt.

```
expression
     : { object }
     ¦ { slot { expression } }
     ¦ { method { expression } }
     ¦ operator_expression
```

Im <object>-Zweig terminiert die Beschreibung durch Nennung des Zeitreihenobjekts, aus dem sich das beschriebene Zeitreihenobjekt errechnet.

```
object
     : ( name level typ unit periods )
```

Mit <slot> kann die Länge der Werte durch Eingrenzung des zeitlichen Gültigkeitsbereichs eingeengt werden.

Der einfachste Fall einer expression ist eine 1:1-Abbildung, d.h. die Kopie eines anderen (elementaren oder höheren) Zeitreihenobjekts. Dabei müssen, wie am folgenden Beispiel ersichtlich, die identifizierenden Namen unterschiedlich und der Level des neuen Objekts höher sein. Das folgende Beispiel zeigt die Berechnung der Gesamtpersonalausgaben für das Lager Bamberg aus den Ausgaben für den einzigen Mitarbeiter "Krause."

```
NODE_DESCR (Angestellte 1 value DM MONTH)
     GLOB_MODEL Lager_BA
     STRUCT_MODEL Ausgaben
 { (Krause 0 value DM MONTH) }
```

### 7. 3. Beschreibung höherer Zeitreihenobjekte durch Verknüpfung von Zeitreihenobjekten

Unter diesem, in der Beschreibungssprachendefinition <operator_expression> genannten Konstrukt versteht man die Verknüpfung von mindestens zwei Zeitreihenobjekten zu einem neuen, übergeordneten höheren Zeitreihenobjekt. Am Beispiel von Abb. 6 entstehen die Werte des Zeitreihenobjekts "Angestellte" durch Aufsummierung der entsprechenden Werte der Zeitreihenobjekte "Maier", "Krause" und "Lehner".

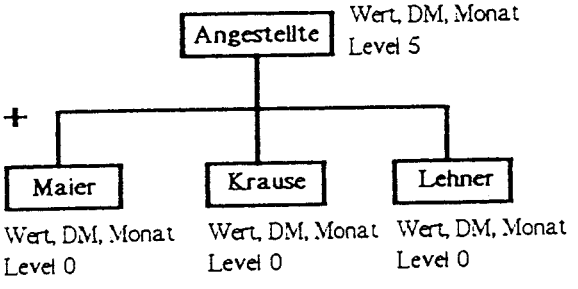



Abb. 6: Additive Verknüpfung von Zeitreihenobjekten

Diese Verknüpfung ist so realisiert, daß im Anweisungsteil, d.h. nach der ersten "{", mit dem Operator (z.B. "+") zwei Zeitreihenobjekte verknüpft werden. Der Operator beschreibt die Verknüpfung (z.B. Addition) zweier Zeitreihen-Objekte, eines Zeitreihenobjekts mit einer Zahl oder mit zwei Zahlen. Werden zwei Zahlen miteinander verknüpft, so ist das Ergebnis wieder eine Zahl. Wird ein Zeitreihenobjekt mit einer Zahl verknüpft, so ist das Ergebnis ein Zeitreihenobjekt und jeder Wert wird mit der Zahl verknüpft. Werden zwei Zeitreihenobjekte miteinander verknüpft, so müssen sie die gleiche Periodizität besitzen.

Bei normalen Operatoren, z.B. "+", wird zu einem Datum nur dann ein Ergebnis errechnet, wenn beide Ausgangs-Zeitreihenobjekte zu diesem Datum einen Wert besitzen. Total-Operatoren (z.B. "T+") errechnen zu jedem Datum, zu dem in mindestens einem Zeitreihenobjekt ein Wert eingetragen ist, einen Wert.

Somit ergibt sich die folgende formale Definition von Bildungsregeln mit Operatoren:

```
operator_expression
     : int_expression + int_expression
     ¦ int_expression - int_expression
     ¦ int_expression * int_expression
     ¦ int_expression / int_expression
     ¦ int_expression "MOD" int_expression
     ¦ int_expression "DIV" int_expression
     ¦ int_expression T+ int_expression
     ¦ int_expression T- int_expression
     ¦ int_expression T* int_expression
     ¦ int_expression T/ int_expression
     ¦ int_expression "TMOD" int_expression
     ¦ int_expression "TDIV" int_expression
     ¦ expression <+> expression

int_expression
     : NUMERIC
     ¦ expression
```

Operatoren sind "links-nach-rechts"-assoziativ, so daß z.B. die Addition mehrerer Zeitreihenobjekte auf die zweier Zeitreihenobjekte zurückgeführt werden kann. Sie verhalten sich auch sonst wie ihre Gegenstücke aus der Arithmetik. Folgendes Beispiel zeigt die totale Addition der Gehälter der Angestellten "Maier", "Krause" und "Lehner" zum Gesamtposten "Angestellte".

```
NODE_DESCR (Angestellte 10 value DM MONTH)
     GLOB_MODEL Lager_BA
     STRUCT_MODEL Ausgaben
     { (Maier 0 v alue DM MONTH) } +
     { (Krause 0 value DM MONTH) } +
     { (Lehner 0 value DM MONTH) }
```

Anstelle eines Zeitreihenobjekts (entspricht in der Grammatik einer expression) kann auch eine Zahl stehen. Diese wird zu allen Werten des Zeitreihenobjekts addiert, wie im folgenden Beispiel die Zahl 10 zu allen Werten des Zeitreihenobjekts Lehner:

```
NODE_DESCR (Angestellte 10 value DM MONTH)
     GLOB_MODEL Lager_BA
     STRUCT_MODEL Ausgaben
     { (Lehner 0 value DM MONTH) } + 10
```

Ändert sich die Berechnungsvorschrift für Kenngrößen (z.B. durch eine Gesetzesänderung), so ist die Handhabung von Strukturbrüchen erforderlich. Die Beschreibungssprache kann sie als <operator_expression> ausdrücken und stellt dafür den Operator <+> zur Verfügung:

```
NODE_DESCR object
expression_1 <+> expression_2
```

Dabei wird das zu berechnende Zeitreihenobjekt mittels zweier verschiedener Berechnungsvorschriften (expression_1 und expression_2) errechnet. Dabei werden zuerst alle Werte, die mit der Berechnungsvorschrift expression_1 errechenbar sind, berechnet. Dann folgt Analoges für expression_2. Zuletzt wird geprüft, daß sich im Sinne der Zeitachse der letzte

(durch expression_1 errechnete) Wert direkt vor dem ersten (durch expression_2 errechneten) Wert befindet. Ggf. muß, um keine Fehlermeldung zu erzeugen, der Zeitraum für eine oder beide expressions eingeschränkt werden. Der Operator <+> eignet sich auch zur Darstellung externer Ereignisse bei Simulationen.

## 7. 4. Beschreibung eines höheren Zeitreihenobjekts unter Benutzung von Umformungsregeln (Methoden)

Bei diesem Verfahren werden, wie in Abb. 7 dargestellt, die Werte eines höheren Zeitreihenobjekts aus einem einzigen anderen Zeitreihenobjekt (d.h. systemtechnisch gesehen aus einem elementaren oder höherem Zeitreihenobjekt mit niedrigerem Level) berechnet.

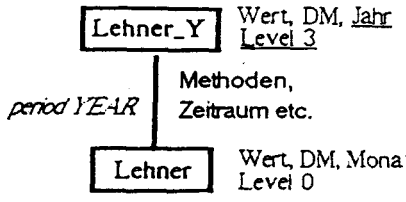



Abb. 7: Bildung eines höheren Zeitreihenobjekts durch die Anwendung von Methoden (im Sinne einer Abbildung)

Zur Umformung der Werte des Zeitreihenobjets können statistische Verfahren, eine Änderung der Periodizität und/oder eine Einschränkung des Zeitraums herangezogen werden. An statistischen Verfahren stehen derzeit die Glättung durch gleitende Durchschnitte, die Indizierung, die Versetzung von Zeitreihen (z.B. zur Ermittlung des Vorjahreswertes) zur Verfügung. Sie können jederzeit erweitert werden. Die Syntaxdefinition der Beschreibungssprache für Sprachkonstrukte mit Anwendungen von Methoden lautet:

```
method
     : sub_method
     | method sub_method

sub_method
     : "period" period_param
     | "period" period_param per_method
     | "unit" unit_param
     | "per_average" INTEGER
     | "heur_per_average" + INTEGER
     | "per_average" INTEGER
     | "heur_per_average" + INTEGER
     | "shift" INTEGER period_param
     | "shift" + INTEGER period_param
     | "shift" - INTEGER period_param
     | "index" date
     | "lin_trend"
     | "lin_trend" slot
     | "exp_trend"
     | "exp_trend" slot
```

Wie aus der Syntaxdefinition für <expressions> (s. Kap. 7.2.) und den Beispielen für die einzelnen Verfahren ersichtlich, steht der Methodenaufruf hier (zusammen mit ggf. benötigten Parametern) vor der expression, auf die er angewendet werden soll.

Ein einfaches Beispiel für Methoden ist die Änderung der Periodizität. Das nachfolgende Beispiel zeigt die Umrechnung des Gehalts des Angestellten Lehner von monatlichen in jährliche Werte.

```
NODE_DESCR (Lehner_Y 3 value TDM YEAR)
   GLOB_MODEL Ausgaben
   STRUCT_MODEL Lager
{ period YEAR (Lehner 0 value DM MONTH) }
```

Wird keine Rechenvorschrift <per_method> angegeben, so werden bei der Umrechnung von kleiner auf große Einheit wie im obigen Beispiel die Werte aufsummiert; bei einem Zeitreihenobjekt vom Typ "index" wird das Mittel genommen. In der umgekehrten Richtung wird der entsprechende Bruchteil (z.B. 1/12) gebildet (und beim Typ-Index der Wert unverändert gelassen). Als Rechenvorschriften stehen Mittelwerte, Differenz max-min im jeweiligen Zeitraum, maximaler Wert, minimaler Wert etc. zur Verfügung. Andere Methoden sind gleitender Durchschnitt (statistisch und heuristisch) zum Ausschalten zeitlicher Schwankungen, die Umrechnung von Einheiten sowie die Indexbildung (z.B. 1.1980 = 100).

Das Versetzen von Zeitreihen auf der Zeitachse kann z.B. zu Zeitvergleichen mit Vergangenheitsdaten und Zukunftsprognosen verwendet werden. Dem Methodennamen "shift" folgen als Parameter eine INTEGER-Zahl und der Typ der Zeiteinheit. Die INTEGER-Zahl ist die Anzahl der Zeiteinheiten, um die verschoben werden soll. Die Werte für Zeiteinheiten (period_param) entsprechen den Angaben für die Periode aus den Parametern der Zeitreihenobjektdefinition (DAY , .. ,YEAR). Zusätzlich dazu kommt die Wertebelegung "STEP". STEP wird intern durch die Periodizität der aktuellen expression substituiert.

Das folgende Beispiel erzeugt die Vorjahreswerte:

```
NODE_DESCR (Lehner_vj 10 value DM MONTH)
     GLOB_MODEL Lager_BA
     STRUCT_MODEL Ausgaben
     { shift -12 STEP (Lehner 0 value DM MONTH) }
```

## 7. 5. Kombination verschiedener Beschreibungstypen

Die in den vorstehenden Abschnitten dargestellten Techniken lassen sich, wie in Abb. 8 dargestellt, beliebig verknüpfen:

Die Beschreibungssprache hierzu:

```
NODE_DESCR (Lagerkosten 12 index .1990 YEAR)
   GLOB_MODEL Lager_BA
   STRUCT_MODEL Ausgaben
{ index .1990 lin_trend L_DATE .2000
 { { period YEAR (Maier 0 value DM MONTH) } +
   { period YEAR (Krause 0 value DM MONTH) } +
   { period YEAR (Aushilfen 0 value DM DAY) } +
   { period YEAR (Material 0 value DM QUARTERYEAR) }
 }
}
```

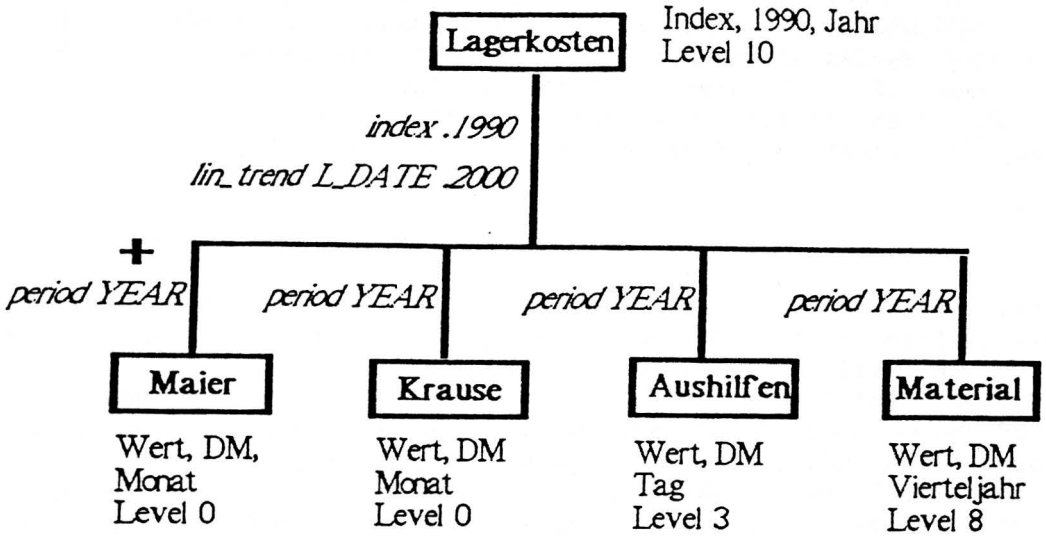



Abb. 8: Kombination von Verknüpfungen und Methoden

Expressions können sich demnach gegenseitig beliebig oft (rekursiv) aufrufen. Dadurch lassen sich Hilfs-Zeitreihenobjekte ohne eigenständige Semantik vermeiden (vgl. hierzu die Vielzahl unnötiger Hilfskennzahlen in den klassischen Kennzahlenmodellen [Lef 76]).Dies erhöht die Übersichtlichkeit für den informationssuchenden Benutzer.

## 7. 6. Beschreibungssprache und Strukturmodelle

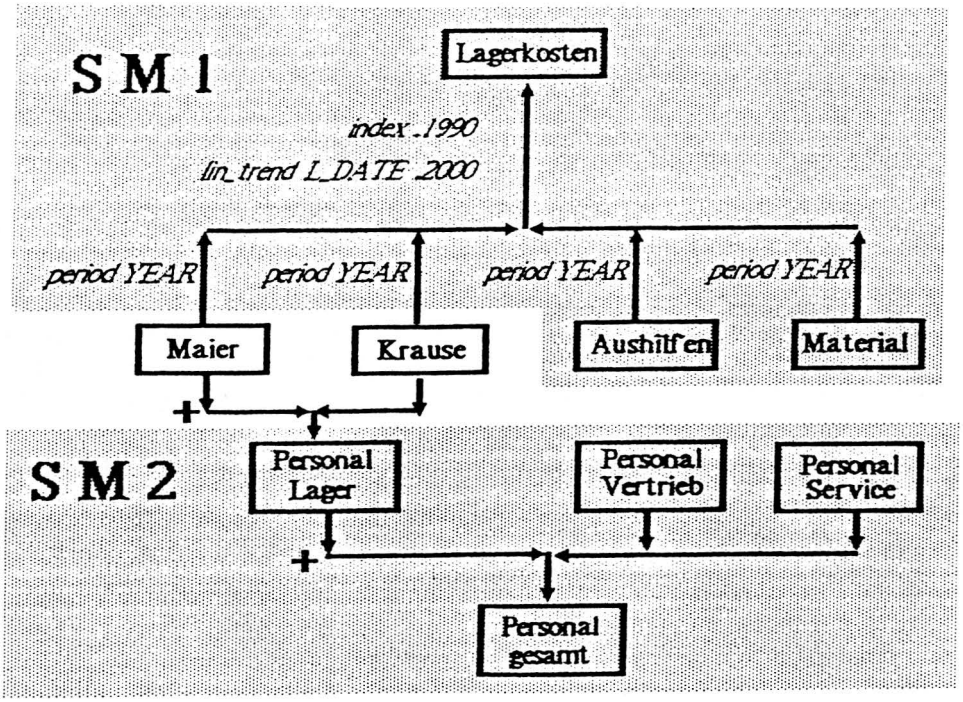



Abb. 9: Zwei Strukturmodelle über elementaren Zeitreihenobjekten

Abb. 9 zeigt die Anwendung der Beschreibungssprache bei der Definition von Strukturmodellen. Dargestellt sind die zwei elementaren Zeitreihenobjekte Maier und Krause. Auf diesen und anderen, aus Platzgründen nicht dargestellten elementaren Zeitreihenobjekten fußen die Strukturmodelle SM1 und SM2. Diese sind für unterschiedlichste Zwecke konstruiert. SM1 ist eine einfache Aufsummierung der Personalkosten nach Organisationseinheiten. SM2 bezweckt eine Prognose der Kostenenentwicklung eines Lagers. Die höheren Zeitreihenobjekte ohne "Unterbau" (z.B. Aushilfen, Personal, Vertrieb) errechnen sich entweder aus nicht dargestellten elementaren Zeitreihenobjekten oder aus höheren Zeitreihenobjekten eines anderen Strukturmodells.

## 8. Implementierung in Stichpunkten

Die Implementierung verfolgt das Prinzip einer netzwerkfähigen Client-Server-Architektur. Es können gleichzeitig beliebig viele Serverprozesse in einem Netz betrieben werden. Als Beispiele sind derzeit Server für Bilanzkennzahlen und für Kennzahlen aus der Materialwirtschaft implementiert. Der Server ist größtenteils in Prolog geschrieben. Jeder Server ist ein Prozeß. Der Client für die graphische Oberfläche ist in C unter X Window implementiert. Clients greifen über Netz auf die Server zu, transformieren die von den Servern gelieferten Informationen in eine für den Anwender lesbare Form, erlauben die Eingabe zusätzlicher Daten und stellen einen Editor zur Erstellung der Beschreibungssprachen bereit. Die Kommunikation wird über ein eigenentwickeltes Protokoll abgewickelt. Darunter liegt eine Kommunikation über ETHERNET und stream sockets.

## 9. Stand des Projekts

Derzeit sind implementiert:

* Die netzwerkfähige Verwaltung der Zeitreihenobjekte,
* ein graphikfähiger Client zur Interaktion mit Endbenutzern,
* Compiler für Bildungsregeln und für die hier nur kurz vorgestellten Demons.

Die Rechnerumgebung ist UNIX-System V und SunOS. Der Client für den Endbenutzer arbeitet unter X Window. Die Implementierung der Textgeneratoren und der Inferenzmaschine ist einem Folgeprojekt vorbehalten. Die Demonsprache und das Implementierungskonzept werden in späteren Berichten im Detail beschrieben werden.

## Literatur:

American Management Association, 1960
Executive Commitee Control Charts. A description of the Du Pont chart system for appraising operating performance; in: AMA Management Bulletin Nr. 6

Augsburger W., Rieder H., Schwab J., 1989
Aufbereitung numerischer Informationen für des Top- und mittlere Management; in: F. Roithmayr (Hrsg.); 9. Symposium "Der Computer als Instrument der Forschung und Lehre in den Sozial- und Wirtschaftswissenschaften", Innsbruck; Oldenbourg, Wien

BIFOA, 1980
Kennzahlenhandbuch der Materialwirtschaft, BIFOA Arbeitspapier 80AO3; Köln

Bläser, P., 1978
Betriebswirtschaftliche Kennzahlen, Teil 4: Materialwirtschaftliche Kennzahlen; in: BKK 23 (1978); Gebertshain

Bratko, I., 1987
PROLOG: Programmierung für künstliche Intelligenz; Wiesbaden

Frei, H. P., Jauslin J.-F., 1983
Graphical Presentation of Information and Services: A User Oriented Interface; in: Information Technology: Research and Development Vol. 2, pp. 23-42

Herczeg M., 1986
Eine objektorientierte Architektur für wissensbasierte Benutzerschnittstellen; Dissertation am Institut für Informatik der Universität Stuttgart

Kuhlen R., Hammwöhner R., Sonnenberger G., Thiel U., Yetim F. 1989
TWRM-TOPOGRAPHIC, Abschlußbericht; Universität Konstanz, Informationswissenschaft, Forschungsberichte, Bericht TOPOGRAPHIC 15/89

Kupfer W., 1982
Konzeption und Programmierung eines Kennzahlensystems als Teil des computergestützten Informations- und Planungswesens in einem Energieversorgungsunternehmen; unveröffentlichte Diplomarbeit am Lehrstuhl Prof. Mertens der Friedrich-Alexander-Universität Erlangen-Nürnberg

Leffson U., 1977
Bilanzanalyse; Stuttgart

Meyer C., 1976
Kennzahlen und Kennzahlensysteme; Stuttgart

Nonhoff J., 1989
Entwicklung eines Expertensytems für das DV-Controlling; Springer Berlin et al

Oehler O., 1978
Checklist Frühwarnsystem mit Alarmkennziffern

Rehkugler H., Poddig T.,1988
Bilanzanalyse; Oldenbourg München, Wien

Schäfer Th., Krug P., 1988
Stand des Expertensystems MEVEX zur Analyse von fünf aufeinader folgenden Jahresabschlüssen; Universität Erlangen-Nürnberg, Arbeitspapiere Informatik-Forschungsgrupe VIII

Staud J., 1985
Online Retrieval in Numeric Data Bases; IATUL Proceedings 17

Zaiß Peter, 1988
Einführung eines Informationssystems zur Steuerung und Kontrolle der Materialwirtschaft eines Energieversorgungsunternehmens; Unveröffentlichte Diplomarbeit am Lehrstuhl Prof. Oechsler der Universität Bamberg