

Systemtheoretische Repräsentation von Strukturen und Bewertungsfunktionen über zeitabhängigen betrieblichen numerischen Daten

Prof. Dr. Walter Augsburger, Bamberg

Dipl. Inf. Wiss. Klaus Helge Rieder, Bamberg

Dipl. Kfm. Hans-Jürgen Schwab, Bamberg

Zusammenfassung: Die Aufbereitung, Beobachtung und Bewertung des zeitlichen Verlaufes einer Vielzahl von Kennzahlenwerten ist für die Vorbereitung mannigfaltiger unternehmerischer Entscheidungen hilfreich. Viele der dabei anfallenden Tätigkeiten können durch den Einsatz moderner Informatikmethoden und Workstations effizient unterstützt werden.

Dies setzt jedoch eine formale Spezifikation von Objekten und Strukturen voraus, auf die der Anwender betriebliche Sachverhalte abbilden kann.

In diesem Tagungsbeitrag wird eine Spezifikation vorgestellt. Sie ist Grundlage des im Projekt "EISREVVU"¹ entwickelten Softwaresystemes, das den Endanwender bei der individuellen Modellierung und Verwaltung von ablauffähigen Modellen, die dieser Spezifikation genügen, unterstützt.

Abstract: For preparing different kinds of management decisions it is useful to prepare, to watch and to validate the behaviour of many time series. A great deal of work involved can be supported by modern methods of computer science and workstations.

This requires a formal specification of objects and structures onto which the user can map his business's structures.

This paper presents the specification for the "EISREVVU" software. It supports the user in individually modelling and administrating models according to this specification.

1 Einführung in die Problemstellung

Numerische Daten in Form von Kennzahlen werden in Unternehmen häufig als Entscheidungshilfe herangezogen. Die Kennzahlen sowie die Regeln für deren Bewertung unterscheiden sich dabei von Entscheidungsträger zu Entscheidungsträger.

Aus dieser einfachen Erkenntnis heraus entstand die Motivation, ein flexibles Modellierungssystem für Kennzahlensysteme und Bewertungsfunktionen für Folgen von Kennzahlenwerten zu entwickeln. Neben der Modellierungsmöglichkeit besitzt dieses Softwaresystem auch die Fähigkeit, die vom Anwender entwickelten Modelle zu verwalten und die Kennzahlenwerte automatisch zu überwachen.

¹Das Projekt "EISREVVU" (Entscheidungsunterstützendes Informationssystem für Energieversorgungsunternehmen) wird finanziell und fachlich durch die Unternehmen Mainkraftwerke AG und Energieversorgung Oberfranken AG gefördert.

Das Softwaresystem "EISREVVU" ist auf grafikfähigen, vernetzten Workstations unter UNIX V.3 und BSD 4.3 als Mehrprozesssystem mit Prolog und C implementiert.

In diesem Tagungsbeitrag werden (in vereinfachter Form) die Strukturen und die Objekttypen vorgestellt², auf die ein Anwender betriebliche Sachverhalte aus seinem Unternehmen bzw. der Unternehmensumwelt abbilden kann. Nicht eingegangen wird aus Platzgründen in unserer Modelldarstellung

1. auf Strukturen über Dämonen, die aus Gründen der besseren Transparenz für den Benutzer eingeführt wurden,
2. auf die Organisation der Modelle in größeren Einheiten mit Hilfe von Obersystemen,
3. auf die Zugriffsrechte einzelner Benutzer mit Hilfe von Zuordnungsrelationen und
4. die Möglichkeiten zur Verwaltung zusätzlicher textueller Information.

Unter dem Begriff "Modell" verstehen wir die Repräsentation von Systemen mit Relationen über Kennzahlen und Bewertungsfunktionen. Eine formale Definition unseres Systembegriffs "Modell" erfolgt in Abschnitt 2.1.

Wir verstehen unter einem Objekt³ ein Tupel, dessen Komponenten Elemente von Mengen bzw. Mengen, Relationen sowie andere hierarchisch tieferstehende Objekte sein können.

In der hier vorgestellten systemtheoretischen Spezifikation werden zunächst auf einer

1. "globalen Ebene" die Strukturen über den Objekten unterschiedlichen Typs (unter Vernachlässigung der Ausprägung der strukturellen Beziehungen zwischen diesen und dem inneren Aufbau der Objekte) und auf einer
2. "lokalen Ebene" der innere Aufbau der Objekte mit den o.g. Komponenten dargestellt und in Abschnitt 4 ein
3. Beispiel für Operationen, die zu einem sinnvollen Umgang mit den Objekten und Strukturen notwendig sind, gebracht.

Dieses 2 Ebenenkonzept wurde auch konsequent bei der Implementation umgesetzt. Der Anwender kann zwischen einer globalen Sicht auf das vom ihm entwickelte Kennzahlenmodell und Editoren wechseln, in denen er lokal die innere Struktur seiner Objekte festlegen kann. Bei den ersten Pilotanwendungen hat sich dieses 2 Ebenenkonzept als sehr nützlich erwiesen.

2 Beschreibung der globalen Strukturen

2.1 Modelle

Die Modelle bilden die formale abstrakte Repräsentation der Objekte, der Struktur von Objekten und der Relationen zwischen den Objektmengen, die ein Anwender selbst definiert. Ein Modell $m \in M$ ist ein

²Zur Notation und relationenalgebraischen Fachbegriffen s. [3].

³Der hier verwendete Objektbegriff ist also eher mit dem Elementbegriff der allgemeinen Systemtheorie verwandt als mit dem Objektbegriff, der in den objektorientierten Programmiersprachen anzutreffen ist.

System:

$$m = (n_{mo}, TM, O, \Gamma_{mo}).$$

Ein Modell besteht aus

- einem eindeutig identifizierenden Namen n_{mo} , der als Schlüssel dient,
- einer Menge von Teilmodellen TM ,
- einer Menge O von Objekten unterschiedlichen Typs deren Elemente aus
 - einer Menge von Zeitreihenobjekten Z ,
 - einer Menge von Dämonen D ,
 - einer Menge von Klassen K (in die die Zeitreihenobjekte, die ein Dämon überwachen kann, eingeteilt werden)

stammen $O = Z \cup D \cup K$;

- einer Relation $\Gamma_{mo} \subset (D \cup K \cup Z_H) \times (K \cup Z)$, die die Abhängigkeiten der Objekte beschreibt.

Die Relation Γ_{mo} setzt sich zusammen aus

- einer Relation $\Gamma_{de} \subset D \times (Z \cup K)$, die die Abhängigkeiten der Dämonen D von den Zeitreihenobjekten Z und den Klassen K beschreibt,
- einer Relation $\Gamma_{kl} \subset K \times (Z \cup K)$, die die Abhängigkeiten der Klassen K von Klassen K und Zeitreihenobjekten Z beschreibt,
- einer Relation $\Gamma_{zt} \subset Z_H \times (Z_H \cup Z_E)$, die die Abhängigkeiten der höheren Zeitreihenobjekte Z_H von höheren Z_H und elementaren Zeitreihenobjekten Z_E beschreibt. Damit wird ausgedrückt, daß Zeitreihenwerte höherer Zeitreihenobjekte $z_H \in Z_H$ aus den Zeitreihenwerten anderer Zeitreihenobjekte berechnet werden.

Die Relation Γ_{mo} kann durch folgende wichtige Bedingungen charakterisiert werden:

1. Asymmetrie, d.h. die Transponierte von $\Gamma_{mo} : \Gamma_{mo}^T$ ist Teilmenge von $\overline{\Gamma_{mo}}$.
2. Nichtexistenz von zyklischen Abhängigkeiten zwischen den Objekten: $\Gamma_{mo}^+ \subset \bar{I}$ wobei Γ_{mo}^+ die transitive Abhängigkeit der Objekte beschreibt⁴. Damit ist automatisch auch
3. Irreflexivität von $\Gamma_{mo} : \Gamma_{mo} \subset \bar{I}$ gegeben.

Die Abhängigkeiten zwischen den Objekttypen lassen sich als System $(\{Z_E, Z_H, K, D\}, R)$ veranschaulichen. Die Relation R , als Matrix dargestellt, trägt an den Stellen eine "1", wenn zwischen den Objekttypen eine funktionale Abhängigkeit herrscht, andernfalls eine "0".

⁴Der Operator + bildet die transitive Hülle einer Relation.

	Z_E	Z_H	K	D
Z_E	0	0	0	0
Z_H	1	1	0	0
K	1	1	1	0
D	1	1	1	0

2.2 Teilmodelle

Mit Hilfe von Teilmodellen kann der Anwender seine Modelle strukturieren und nach sachlogischen Gesichtspunkten übersichtlich gruppieren. In Teilmodellen wird also numerische Information aufbereitet und verdichtet. Ein Teilmodell $tm \in TM$ ist durch folgende Attribute charakterisiert:

- einen Namen n_{tmo} , mit dem es innerhalb des Modells identifiziert wird,
- einer Menge von Objekten O_i , die ihrerseits aus
 - einer Menge von Zeitreihenobjekten $Z_i \subset Z$,
 - einer Menge von Klassen $K_i \subset K$ und
 - einer Menge Dämonen $D_i \subset D$

besteht,

- eine Relation $\Gamma_{tmo} \subset (D_i \cup K_i \cup Z_i) \times (K_i \subset Z_i)$, die die Abhängigkeiten zwischen den Objekten beschreibt:

$$tm = (n_{tmo}, O_i, \Gamma_{tmo}).$$

Die Abb. 1 stellt die bis jetzt dargestellte Sichtweise noch einmal graphisch dar.

3 Beschreibung der lokalen Objektstrukturen

3.1 Zeitreihenobjekte

Wir unterscheiden zwischen der Menge der elementaren Zeitreihenobjekte⁵ Z_E und der Menge der höheren Zeitreihenobjekte Z_H . Die elementaren Zeitreihenobjekte stellen in den Modellen diejenigen "Kennzahlen" dar, die direkt aus den operativen Systemen stammen. Damit handelt es sich bei elementaren Zeitreihenobjekten, bildlich gesprochen, um die Kennzahlen, die den Fuß der "Kennzahlenpyramiden"⁶ bilden, in denen die Kennzahlen angeordnet werden. In den Teilmodellen werden Zeitreihenobjekte $z \in Z_E \cup Z_H$ mittels einer vom Endbenutzer definierten Vorschrift $bv : z^k \rightarrow z_H, k \in \mathbb{N}$ zu höheren Zeitreihenobjekten $z_H \in Z_H$ verknüpft. Es gilt: $Z_E \cap Z_H = \emptyset$ und $Z = Z_E \cup Z_H$. Zeitreihenobjekte haben folgende Attribute:

⁵Ein genauerer Überblick über die Zeitreihenobjekte wird in [2] gegeben.

⁶Modelle oder Teilmodelle müssen keine Pyramidenform besitzen.

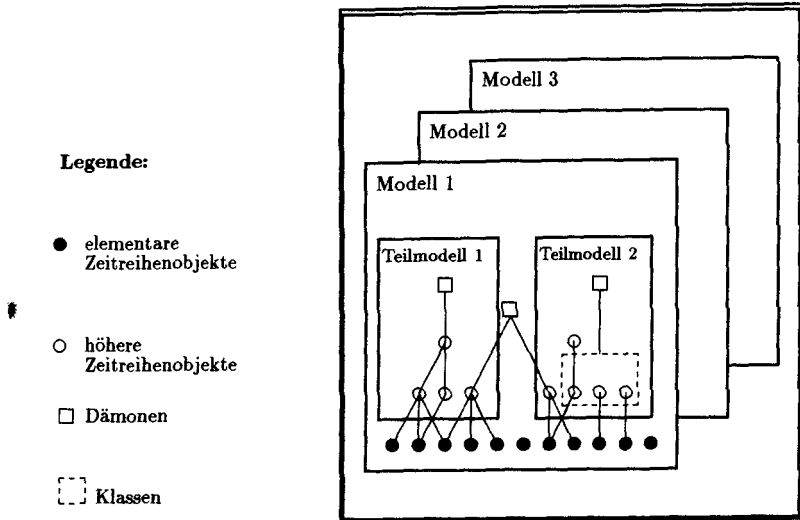


Abbildung 1: Objekte und Relationen von Modellen

- Eine Einheit ϵ , deren Ausprägung aus einer vorgegebenen Menge $E = \{dm, kg, to \dots\}$ stammen muß.
- Einen Typ y , der sich auf die Zeitreihe bezieht und der aussagt, ob es sich bei den Zeitreihenwerten entweder um normale Werte (value), um Indexzahlen (index), um Prozentwerte (percentage) oder um errechnete Werte (relation), also beispielsweise Mitarbeiter pro Betriebseinheit, handelt.
- Ein weiteres Attribut der Zeitreihenobjekte ist die Periodizität $p \in P$, die ebenfalls nur vorgegebene Ausprägungen annehmen darf. $P = \{Jahr, Halbjahr, Vierteljahr, Monat, Tag\}$.
- Alle Zeitreihenobjekte tragen einen Namen n aus der Menge der für Zeitreihenobjekte zulässigen Namen NO_{zr} , der innerhalb eines Modells für eindeutige Identifikation sorgt.
- Jedes Zeitreihenobjekt besitzt eine Zeitreihe ts , die aus Tupeln mit Datumsangabe und Zeitreihenwert der Form (d_i, z_i) besteht. An die Zeitreihe ts können beispielsweise neue Elemente angehängt werden. Bei elementaren Zeitreihenobjekten geschieht dies durch einen "Update" mit Zeitreihenwerten aus operativen Systemen, bei höheren Zeitreihenobjekten werden neue Elemente durch Berechnung erzeugt.
- Die höheren Zeitreihenobjekte werden zusätzlich durch eine Berechnungsvorschrift bv aus der Menge aller gültigen Berechnungsvorschriften BV beschrieben. Diese gibt an, wie sich die Zeitreihe berechnet. Diese Berechnungsvorschrift ist bei elementaren Zeitreihenobjekten als ein leeres Wort vorstellbar. Die Syntax und Semantik des Kalküls der Berechnungsvorschrift ist in [1] dargestellt.
- Die Zeitreihenwerte der höheren Zeitreihenobjekte werden aus den Zeitreihenwerten anderer Zeitreihenobjekte $Z_{zr} \subset Z$ direkt berechnet. Die Menge der Z_{zr} wird zur verallgemeinerten Darstellung eines

Zeitreihenobjektes bei Z_E als leer aufgefaßt.

Ein Zeitreihenobjekt $z \in Z$ kann als ein Tupel aus einer Einheit e , einem Typ y , einer Periodizität p , einem Namen n , einer Zeitreihe ts , einer Beschreibungssprache bv und einer Menge von Z_{zr} , aus deren Zeitreihen seine Zeitreihe berechnet wird, beschrieben werden:

$$z = (n, e, y, p, ts, bv, Z_{zr}).$$

3.2 Dämonen

Weitere Objekte innerhalb eines Modells sind Bewertungsfunktionen für die Werte einer Zeitreihe von Zeitreihenobjekten, die "Dämon" genannt werden. Dämonen untersuchen die Zeitreihenwerte von Zeitreihenobjekten auf vorgegebene Muster hin und erstellen, falls diese Muster vorliegen, Meldungen. Dabei können die Dämonen so eingestellt werden, daß sie automatisch, ohne weiteres Zutun des Benutzers dann aktiv werden, wenn sich die Zeitreihen von Zeitreihenobjekten verändern.

"Muster" innerhalb einer Zeitreihe können beispielsweise kurze periodische Schwankungen, Trends oder einfach das Über- oder Unterschreiten vorgegebener Werte sein. Die Meldungen können neben einfachen Texten auch statistische Informationen usw. umfassen. Die Dämonen haben folgende Attribute:

- Sie haben einen Namen $n \in N_{de}$, der als Schlüssel dient.
- Jedem Dämon wird durch den Anwender eine Menge Zeitreihenobjekte $Z_{de} \subset Z$ zugeordnet, die der Dämon überwacht. Diese Zuordnung kann sowohl durch die Angabe der Menge der Z_{de} oder durch die Angabe einer Klasse k_{de} (s. Abschnitt 3.3) beschrieben werden.
- Die Dämonen werden abhängig von bestimmten Ereignistypen $ev \in \{SELECTION, UPDATE, CHANGE, ALWAYS\}$ aktiv. Wenn die Wertreihe eines Zeitreihenobjektes $z \in Z_{de}$, das dem Dämon zugeordnet ist, verändert wird, löst dies ein UPDATE-Ereignis aus. Wenn die Beschreibung eines höheren Zeitreihenobjektes $z_H \in Z_{de}$, an das der Dämon gebunden ist, verändert wird, löst dies ein CHANGE-Ereignis aus. Wenn der Benutzer den Dämon per Hand aktiviert, löst dies ein SELECTION-Ereignis aus. Alle anderen Ereignisse, die bewirken können, daß der Dämon die Muster der zugeordneten Zeitreihenobjekte Z_{de} als zutreffend erkennt, sind in der Restkategorie ALWAYS eingeordnet⁷. Die Einführung von Ereignistypen erfolgt letztlich aus Effizienzgründen und soll eine ressourcenschonende Abarbeitung der Dämonen ermöglichen.
- Die Verfahren, mit denen ein Dämon die zugeordneten Zeitreihenobjekte untersucht, heißen "Mustererkennungsverfahren" mf . Die Mustererkennungsverfahren $mf \in MF$ stellen eine Abbildung von Zeitreihenwerten auf die Werte wahr oder falsch dar:

$$mf(ts_1, \dots, ts_n) \rightarrow \{\text{wahr}, \text{falsch}\}.$$

⁷Dämonen lösen selbst keine Berechnungsvorgänge aus.

Ein Dämon kann über eine Menge MF von Mustererkennungsverfahren verfügen. Mustererkennungsverfahren können durch logisches \vee und \wedge , wie im folgenden Beispiel, verknüpft werden:

$$mf_1(ts_1, ts_2, ts_3) \vee mf_2(ts_2, ts_3) \wedge mf_n(ts_4).$$

Dieser logische Ausdruck wird ausgewertet, und falls er den Wahrheitswert "wahr" besitzt, führt der Dämon die zugehörigen Aktionen A aus. Die Aktionen bestehen im wesentlichen in der Berechnung von statistischen Informationen und der Ausgabe von Meldungen MD für die Benutzer.

Ein Dämon d kann formal beschrieben werden als ein Tupel aus einem Namen n , einer Menge von Zeitreihenobjekten Z_{de} , die er beobachtet oder einer Klasse k_{de} , deren Mitglieder der Dämon beobachtet, eines Ereignisstyps ev , auf das hin der Dämon aktiv wird, einer Menge Mustererkennungsverfahren MF , einer Menge Aktionen A und einer Menge von Meldungen MD , die er erzeugt hat:

$$d = (n, Z_{de}, k_{de}, ev, MF, A, MD).$$

3.3 Klassen

Als weitere Objekte existieren in einem Modell noch Klassen K , in die Zeitreihenobjekte eingeteilt werden. Die Definition von Klassen verfolgt das Ziel, Dämonen anstatt an einzelne Zeitreihenobjekte an eine ganze Menge von Zeitreihenobjekten zu binden. Der Dämon überwacht dann nicht einzelne Objekte, sondern alle Klassenmitglieder oder die Mitglieder von (Unter-) Klassen, aus denen eine (Ober-) Klasse besteht. Klassen k haben folgende Attribute:

- einen identifizierenden Namen n_{kl} ,
- eine Menge von Zeitreihenobjekten $Z_{kl} \subset Z$ oder
- eine Menge von Klassen $K_{kl} \subset K$.

Eine Klasse k läßt sich als ein Tupel beschreiben: $k = (n_{kl}, Z_{kl}, K_{kl})$.

Ein Zeitreihenobjekt kann gleichzeitig in mehreren Klassen "Mitglied" sein. Die Klassen selbst können bezüglich ihrer Mitglieder, also Zeitreihenobjekte oder Klassen, "überlappend" sein.

Die Abhängigkeit der Klassen von Zeitreihenobjekten oder Klassen beschreiben wir mit der Relation: $\Gamma_{kl} \subset K \times (Z \cup K)$, wobei gilt: $\Gamma_{kl} \subset \Gamma_{m_0}$.

4 Operationen und Konsistenzbedingungen

In diesem Abschnitt werden die Operationen zum Anlegen von Strukturen, also von Modellen und Teilmodellen und Objekten aus den Mengen der Dämonen D , der Klassen K und der Zeitreihenobjekte Z aufgelistet. Für diese Operation definieren wir einen Operator \cup , mit dem zu einer Menge von Objekten des gleichen Typs bzw. einer Relation aus Tupeln des gleichen Typs ein Objekt oder eine Objektmenge bzw. Tupel des gleichen Typs hinzugefügt werden.

Die Operationen zum Anlegen von Strukturen und Objekten sind unproblematisch. Wir fordern nur und überprüfen auch, daß gleiche Namen nicht doppelt vergeben werden dürfen. Beim Anlegen von neuen Strukturen und Objekten werden die Mengen und die Relationen, in denen die Abhängigkeiten festgehalten sind, verändert. Die Mengen und Relationen, die sich dabei ergeben, sind durch einen Strich gekennzeichnet.

Modelle	M'	$= M \cup \{m_l \mid i \neq j \iff n_{m_o,i} \neq n_{m_o,j} \neq n_{m_o,l}\}$ mit $i, j, l \in \mathbb{N}; j, i = 1, \dots, M ; l = M + 1$
Teilmodelle	TM'	$= TM \cup \{tm_l \mid i \neq j \iff n_{tm_o,i} \neq n_{tm_o,j} \neq n_{tm_o,l}\}$ mit $i, j, l \in \mathbb{N}; j, i = 1, \dots, TM ; l = TM + 1$
Dämonen	D'	$= D \cup \{d_l \mid i \neq j \iff n_{de,i} \neq n_{de,j} \neq n_{de,l}\}$ mit $i, j, l \in \mathbb{N}; i, j = 1, \dots, D ; l = D + 1$
	Γ'_{de}	$= \Gamma_{de} \cup (\{d_l\} \times (Z_{de} \cup \{k_{de}\}))$
	Γ'_{tm_o}	$= \Gamma_{tm_o} \cup (\{d_l\} \times (Z_{de} \cup \{k_{de}\}))$
	Γ'_{m_o}	$= \Gamma_{m_o} \cup (\{d_l\} \times (Z_{de} \cup \{k_{de}\}))$
Klassen	K'	$= K \cup \{k_l \mid i \neq j \iff n_{kl,i} \neq n_{kl,j} \neq n_{kl,l}\}$ mit $j, i, l \in \mathbb{N}; j, i = 1, \dots, K ; l = K + 1$
	Γ'_{kl}	$= \Gamma_{kl} \cup (\{k_l\} \times (Z_{kl} \cup \{K_{kl}\}))$
	Γ'_{tm_o}	$= \Gamma_{tm_o} \cup (\{k_l\} \times (Z_{kl} \cup \{K_{kl}\}))$
	Γ'_{m_o}	$= \Gamma_{m_o} \cup (\{k_l\} \times (Z_{kl} \cup \{K_{kl}\}))$
Zeitreihenobjekte	Z'	$= Z \cup \{z_l \mid i \neq j \iff n_{z_o,i} \neq n_{z_o,j} \neq n_{z_o,l}\}$ mit $i, j, l \in \mathbb{N}; j, i = 1, \dots, Z ; l = Z + 1$
	Γ'_{zr}	$= \Gamma_{zr} \cup (\{z_l\} \times Z_{zr})$
	Γ'_{tm_o}	$= \Gamma_{tm_o} \cup (\{z_l\} \times Z_{zr})$
	Γ'_{m_o}	$= \Gamma_{m_o} \cup (\{z_l\} \times Z_{zr})$

In gleicher Weise sind die Löschooperationen auf Strukturen und Objekten definiert. Sie sind ausführlich in [4] dargestellt.

Literatur

- [1] Augsburger W., Rieder H., Schwab J.
Endbenutzerorientierte Informationsgewinnung aus numerischen Daten am Beispiel von Unternehmenskennzahlen. In: Rainer Kuhlen (Hrsg.), *Pragmatische Aspekte beim Entwurf und Betrieb von Informationssystemen*, Konstanz, Proc. des 1. Internationalen Symposiums für Informationswissenschaft, 241-255, (1990)
- [2] Augsburger W., Rieder H., Schwab J.
Wissensbasiertes, inhaltsorientiertes Retrieval statistischer Daten mit EISREVVU. erscheint in: Workshop Information Retrieval, Darmstadt 24.-25. Juni Proceedings; Springer, Informatik Fachberichte, (1991)
- [3] Schmidt G., Ströhlein T.
Relationen und Graphen. Springer-Verlag, Berlin et al., (1989)
- [4] Schwab J.
Ein computergestütztes Modellierungssystem zur Kennzahlenbewertung. Eingereichte Dissertation, Universität Bamberg, Bamberg, (1991)