

# Datengitter zur grafischen Darstellung von Datenstrukturen

Siegfried Wendt\*

**Stichworte:** Strukturplan, Datenstruktur, Datenbank, Semantik, Software-Dokumentation.

**Zusammenfassung:** Ein Datengitter dient dazu, einen Wert oder den Wertebereich einer strukturierten Datenvariablen als semantischen Verbund von Datenelementen darzustellen. Die Einsetzbarkeit von Datengittern ist nicht auf Datenbankschemata beschränkt; vielmehr wird gerade der Einsatz von Datengittern außerhalb von Datenbankanwendungen besonders betont und am Beispiel eines Grafik-Editors veranschaulicht. Die Datengitter basieren auf den drei erkenntnistheoretisch begründeten Begriffen Objekt, Attribut und Relation. Die grafische Symbolik orientiert sich an der mathematischen Sicht von Mengen und Relationen.

Relation charts for graphic documentation of data structures

**Key-words:** structure chart, data structure, data base, semantics, software-documentation.

**Abstract:** A relation chart is a graphic representation of a value or of the type of a structured data variable in form of a semantic network of data elements. Relation charts cannot only be used to represent schemes in data base systems; the paper rather emphasizes the use of relation charts outside of data base applications. This is illustrated by the example of a graphics editor. Relation charts are based on concepts of cognition theory: object, attribute and relation. The graphic symbols are related to the mathematical view of sets and relations.

## 1 Einleitung

Das Sichtbarmachen unterschiedlicher Struktur Aspekte komplexer Softwaresysteme ist eine grundlegende Voraussetzung für die ingenieurmäßige Beherrschung solcher Systeme. In [1] hat der Autor eine Projektbibliothek auf

\* Prof. Dr.-Ing. Siegfried Wendt, Fachbereich Elektrotechnik der Universität Kaiserslautern, Erwin-Schrödinger-Straße, D-6750 Kaiserslautern

der Basis von Strukturplänen charakterisiert, wobei jedoch keine Vorschläge zur Gestaltung konkreter Strukturpläne gemacht wurden. Ein solcher Vorschlag, der auf jahrelanger Erfahrung mit dem praktischen Einsatz von Strukturplänen bei umfangreichen Softwareprojekten beruht, wird nun mit der vorliegenden Arbeit für den Bereich der Datenstrukturen angeboten. Datenstrukturen bilden eine der drei Klassen, in die die verschiedenen sichtbar zu machenden Struktur Aspekte eingeteilt werden können: Aufbaustrukturen, Ablaufstrukturen und Datenstrukturen. In den Aufbaustrukturplänen werden Systembausteine in einem Kommunikations- und/oder Platzierungsverbund gezeigt. Die Ablaufstrukturpläne zeigen die Verflechtung potentieller Ereignisketten. In einem Datenstrukturplan werden Datenelemente in einem semantischen oder speichertechnischen Verbund gezeigt.

Es fällt auf, daß in Lehrbüchern über Datenstrukturen [2], [3], [4] ausschließlich der speichertechnische Verbund – Felder, Listen, Bäume usw. – behandelt wird, während der semantische Verbund nur als sogenanntes Schema im Zusammenhang mit Datenbanken [5], [6], [7] diskutiert wird. Daraus könnte vorschnell gefolgert werden, daß die Darstellung einer Datenstruktur als semantischer Verbund außerhalb der Datenbanktechnik nicht benötigt werde. Eine solche Folgerung deckt sich jedoch keineswegs mit den Dokumentationsbedürfnissen aus der Praxis. Es gibt viele große Softwaresysteme, die nichts mit Datenbanken zu tun haben, in denen aber doch komplexe Datenstrukturen von zentraler Bedeutung vorkommen. Sowohl für den Entwurf als auch für die spätere Weitergabe der Information über das Produkt ist es unerlässlich, daß die Datenstrukturen dieser Systeme nicht nur hinsichtlich der Adressierungstechnik, sondern auch hinsichtlich ihrer Semantik in Strukturplänen erfaßt werden.

## 2 Beispiel einer Aufgabenstellung

Als Beispiel für ein Softwaresystem, welches sehr stark durch eine komplexe Datenstruktur ohne Bezug zu einer Datenbank geprägt ist, wird der graphische Editor in Bild 1 betrachtet. Er dient als Werkzeug zum interaktiven Gestalten von Strukturplänen für eine Projektbibliothek. Im Hintergrundspeicher werden die Strukturpläne als Textdateien in geeigneter Planbeschreibungssprache abgelegt. Das Einbringen in die Projektbibliothek bleibt hier außer Betracht.

Während der Editiersitzung befindet sich der gerade bearbeitete Plan in einer für die Interaktion geeigneten Codie-

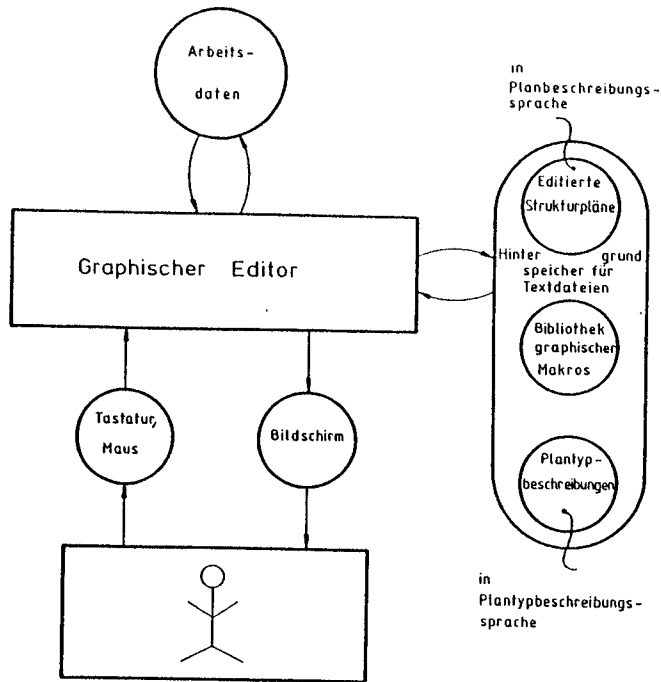


Bild 1 Datenumgebung des Grafik-Editors

rung in Form verketteter Listen als Teil der Arbeitsdaten (s. Bild 1) im Adreßraum des Editorprogramms. Da die Strukturpläne nicht nur als grafische Gebilde, sondern auch als diskrete Strukturen bestimmter Art editiert werden sollen, muß der Editor lesenden Zugriff zu den verschiedenen Plantypbeschreibungen – z.B. Instanzennetz, Petrinetz, Relationsgitter – haben, die ebenfalls als Textdateien im Hintergrundspeicher liegen. Von dort wird die Beschreibung des Typs des gerade zu bearbeitenden Plans geholt und in einen Teil der Datenstruktur der Arbeitsdaten umgewandelt. Damit man häufig vorkommende grafische Symbole nicht bei jedem Auftreten wieder neu aus den grafischen Primitiven aufbauen muß, gibt es die Möglichkeit, solche Symbole in eine Bibliothek grafischer Makros einzubringen, wo sie dann beliebig oft abgerufen werden können.

Aus Umfangsgründen muß hier das System etwas vereinfacht dargestellt werden. So wird hier beispielsweise nicht berücksichtigt, daß im realen System die Arbeitsdaten nicht nur einen einzigen Strukturplan, sondern gleichzeitig mehrere solche Pläne enthalten können, die in verschiedenen Fenstern des Bildschirms dargestellt werden. Aber selbst für den vereinfachten Fall erkennt man leicht, daß das Editationsobjekt in den Arbeitsdaten, welches aus einem zu bearbeitenden Strukturplan samt der zugehörigen Plantypbeschreibung besteht, eine recht komplexe Datenstruktur sein muß. Im Abschnitt 5 wird diese Datenstruktur als Datengitter dargestellt und kommentiert.

### 3 Erkenntnistheoretische Betrachtungen zum Datenbegriff

In DIN 44300 werden Daten definiert als „Zeichen oder kontinuierliche Funktionen, die zum Zweck der Verarbeitung Information aufgrund bekannter oder unterstellter Abmachungen darstellen.“ Diese Definition soll anhand von Bild 2 kurz kommentiert werden: In diesem Bild stehen die Daten in der Mitte und haben zwei Nachbarn, nämlich die physikalische Ausprägung und die Bedeutung. Die Bedeutung ist hierbei identisch mit der in der Norm genannten Information, die bei Kenntnis der Abmachungen den Daten entnehmbar ist. Dieser Vorgang des Entnehmens von Information aufgrund von Abmachungen ist in Bild 2 Interpretation genannt. Da die Daten als Symbolklassen definiert sind, denen die physikalischen Ausprägungen durch Klassifikation zugeordnet werden, sind sie selbst keine Gebilde aus Materie und Energie in Raum und Zeit, sondern nur noch denkbare Gebilde, und sie gehören deshalb bereits in die Welt der Information, selbst wenn ihnen noch keine Bedeutung zugeordnet wurde.

Wenn man dagegen eine Datenstruktur als semantischen Verbund betrachtet, dann sind die Welt der Symbole (s. Bild 2) und damit die Daten im Sinne der Norm völlig uninteressant; es interessiert nur noch die Welt der Bedeutungen. Die grafische Darstellung einer Datenstruktur

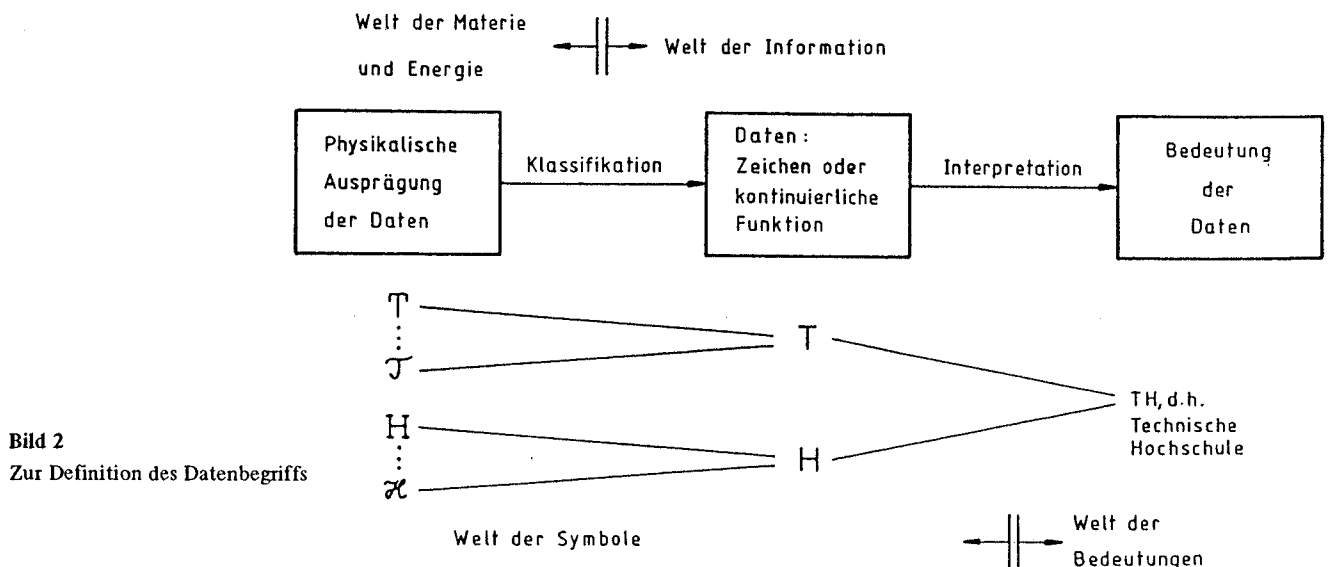


Bild 2 Zur Definition des Datenbegriffs

als semantischer Verbund soll möglichst anschaulich die Elemente und ihre Beziehungen zeigen, die sich „auf natürliche Weise“ beim Nachdenken über die Datenbedeutung „ins Bewußtsein drängen“. Diese Forderung führt zwangsläufig zu Fragen, die der Erkenntnistheorie zuzuordnen sind. Es ist jedoch nicht erforderlich, an dieser Stelle sehr tief in die Erkenntnistheorie einzudringen; als plausible Begründung für die Begriffswelt und Symbolik der Datengitter genügen bereits die folgenden einfachen Betrachtungen.

Daten dienen der Identifikation eines Weltausschnitts, d.h. die den Daten durch Interpretation zuzuordnende Bedeutung kann nichts anderes als ein Hinweis auf einen Teil der erlebten Welt sein. Der Mensch erlebt die Welt in Form von Objekten, die Eigenschaften haben und in Beziehungen zueinander stehen können. Bild 3 zeigt ein Schema, wonach die Objekte nach verschiedenen Kriterien klassifiziert werden können. Außenwelt ist das, was wir durch unsere Sinnesorgane wahrnehmen. Als Objekte der Außenwelt erleben wir sowohl Gegenstände – ein bestimmtes Haus, ein bestimmtes Buch – als auch Prozesse – ein bestimmtes Konzert, eine bestimmte Prüfung. Ein Prozeß ist ein Geschehen, das wir als Einheit erleben. Die Objekte der Außenwelt sind alle konkret, d.h. ihre Existenz ist ohne Bindung an Raum oder Zeit nicht denkbar. Innenwelt ist das, was nur subjektiv erlebbar ist; hierher gehören Gefühle, Gedanken, Vorstellungen und Träume. Als Objekte der Innenwelt erleben wir nicht nur Gegenstände und Prozesse – als Gegenstand eine Glocke, wenn wir Schiller's Gedicht lesen, oder als Prozeß die Angst, die wir in einer bestimmten Gefahrensituation empfinden. Vielmehr erleben wir in der Innenwelt auch noch die abstrakten Objekte, deren Existenz frei von Raum und Zeit gedacht werden kann. Die abstrakten Objekte werden Begriffe genannt. Bevor näher auf die Begriffe eingegangen werden kann, muß zuerst noch etwas über die Eigenschaften der konkreten Objekte und ihre Beziehungen untereinander gesagt werden.

Eigenschaften gehören zum Wesen der konkreten Objekte, sie sind diesen Objekten eigen. Manche Eigenschaften wie beispielsweise Farbe oder Tonhöhe erleben wir ohne Bezug zu einem Referenzobjekt, während andere Eigen-

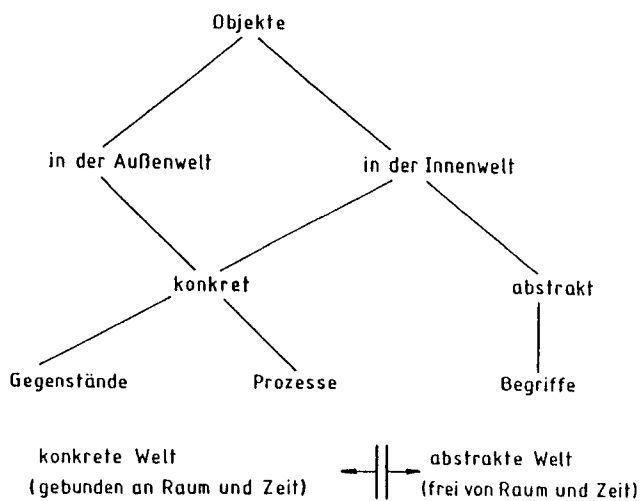


Bild 3 Zum Begriff des Objekts

schaften, nämlich die Positionen eines Gegenstandes im Raum oder eines Prozesses in der Zeit nur unter Bezug zu einem Referenzobjekt erlebbar sind. Dabei erlebt der Betrachter sich selbst als Referenzgegenstand und sein eigenes Erleben als Referenzprozeß.

Während die Erkenntnis der Objektexistenz eine diskrete Welt schafft, wird durch die Erkenntnis der Objekteigenschaften eine kontinuierliche Welt geschaffen. Das Diskrete ist eine Konsequenz der Existenzfrage, während das Kontinuierliche eine Konsequenz des Existenten ist. Ein Grashalm kann entstehen und vergehen, aber solange er existiert, muß er auch eine Länge haben.

Die elementaren Beziehungen, die wir zwischen konkreten Objekten bestehend erleben, sind zum einen Teil Eigenschaftsbeziehungen und zum anderen Teil Existenzbeziehungen. Die Eigenschaftsbeziehungen ergeben sich durch Eigenschaftsvergleich – länger, heller, später –, wobei Positionsbeziehungen wie unter, über oder neben die Betrachterposition mit einbeziehen. Da die Eigenschaftsbeziehungen aus der Kenntnis der Eigenschaften ableitbar sind, bringen sie keine grundsätzlich neue Erkenntnis. Es gibt zwei mögliche Beziehungen zwischen den Existenzen zweier Objekte: Die eine Beziehung ist gegeben, wenn ein Objekt a Bestandteil eines Objekts b ist, wobei a und b entweder zwei Gegenstände oder zwei Prozesse sind. Die andere Beziehung liegt vor, wenn ein Gegenstand a an einem Prozeß b beteiligt ist.

Neben den genannten elementaren Beziehungen zwischen konkreten Objekten kennt man natürlich auch Beziehungen, die nicht mehr so elementar wahrnehmbar sind – beispielsweise sind a und b miteinander verheiratet, oder a ist ein Buch und b sein Autor. Solche Beziehungen sind nicht mehr ohne die Hinzunahme von abstrakten Objekten, d.h. Begriffen erklärbar. Jeder Begriff repräsentiert das Wesen einer Klasse, also das Gemeinsame, was die Elemente einer Klasse kennzeichnet. Begriffe gibt es zu Objekten, zu Eigenschaften und zu Beziehungen. So repräsentieren beispielsweise der Begriff „Konzert“ das Wesen der Klasse aller Konzerte, der Begriff „gelb“ das Wesen der Klasse aller gelben Objekte und der Begriff „zusammenstoßen“ das Wesen der Beteiligung zweier Gegenstände an einem Prozeß aus der Klasse aller Zusammenstöße. Diese Beispiele sind elementare Begriffe, denn die Elemente der zugehörigen Klassen stammen aus der konkreten Welt. Die nichtelementaren Begriffe dagegen repräsentieren jeweils das Wesen einer Klasse von Begriffen. Als Beispiel sei der Begriff „ganze Zahl“ genannt, der das Wesen der Klasse aller ganzen Zahlen repräsentiert, wobei jede einzelne ganze Zahl, also beispielsweise 1985, selbst wieder ein Begriff ist. Jeder Begriff ist ein Objekt, das zu jedem Objekt in der Klasse, deren Wesen er repräsentiert, in einer bestimmten Beziehung steht. Man nennt diese Beziehung Abstraktion, wenn man vom Klassenelement zum Begriff geht, wenn man also die konkrete Welt verläßt oder sich weiter von ihr entfernt, und man spricht von Konkretisierung im umgekehrten Fall. Begriffe sind Objekte ohne Eigenschaften. Wenn man dennoch manchmal zu einem Begriff eine Eigenschaft assoziiert, z.B. eine bestimmte Farbe zum Begriff „gelb“, dann ist dies doch nur eine Eigenschaft eines konkreten Objekts aus der durch den Begriff erfaßten Klasse. Da Begriffe also nicht durch Beschreibung ihrer

Eigenschaften definierbar sind, können sie nur durch Angabe ihrer Beziehungen zu anderen Begriffen und konkreten Objekten gekennzeichnet werden.

Die bis hier angestellten Betrachtungen sollten zeigen, daß die Frage nach der Bedeutung von Daten beantwortet werden kann, indem man die durch die Daten zu identifizierenden Objekte, Eigenschaften und Beziehungen angibt. Diese Erkenntnis bildete auch schon die Basis für das „Entity-Relationship-Modelle“ von Chen [8] und steckt letztlich hinter jedem semantischen Datenmodell [7], [9]. Dies bedeutet aber nicht, daß mit dieser Erkenntnis die Vorgehensweise bei der Darstellung einer Datenstruktur als semantischer Verbund von Datenelementen schon festläge. Daß es hier Variationsmöglichkeiten gibt, sei am Beispiel der Eigenschaftserfassung und der Beziehungsklassifikation aufgezeigt.

Eine Eigenschaft kann man entweder explizit als Beziehung zwischen dem die Eigenschaft habenden Objekt und dem zugehörigen Eigenschaftsbegriff darstellen, oder aber man kann die Eigenschaft als zum Wesen des Objektes gehörend darstellen, wobei dann der Eigenschaftsbegriff und die Beziehung nur noch implizit vorkommen. Der Autor hält die implizite Erfassung für einleuchtender.

Bei der Einteilung der Beziehungen in bestimmte Klassen geht man von der Vorstellung aus, daß es nur eine kleine Anzahl allgemeingültiger Beziehungstypen gebe und daß das Verständnis einer mitzuteilenden Beziehung dadurch gefördert oder gar erst ermöglicht werde, daß die Typenzugehörigkeit angegeben wird. So fällt beispielsweise die Beziehung zwischen den Musikern und ihrem Orchester unter den Typ „Aggregation“, während die Beziehung zwischen einer Sinfonieaufführung, dem Komponisten und dem Orchester eine „Konnexion“ ist [7]. Der Autor kann nicht einsehen, daß eine solche Typisierung das Verständnis der beschriebenen Beziehungen fördert; er erlebt sie vielmehr als unnötige Erschwerung der Kommunikation.

Da es bei jeder Datenbeschreibung darauf ankommt, Objekte, Eigenschaften und Beziehungen mitzuteilen, muß hier noch etwas über die drei grundsätzlich unterschiedlichen Möglichkeiten der Identifikation gesagt werden. Nur die Gegenstände der Außenwelt können direkt identifiziert werden, indem man durch Hinzeigen, Anfassen oder ähnliche Handlungen die Aufmerksamkeit darauf lenkt. Die Möglichkeit der Charakterisierung dagegen besteht für Objekte jeglicher Art; in diesem Fall erfolgt die Identifikation durch Angabe von Eigenschaften und Beziehungen zu anderen Objekten. Auch die Benennung als dritte Art der Identifikation ist für alle Objekttypen möglich. In diesem Fall ist eine Symbolklasse in Form eines leicht reproduzierbaren Musters – für die zwischenmenschliche Kommunikation ein Name – als Stellvertreter des Objektes bekannt, und jedes Auftreten des Namens identifiziert das benannte Objekt.

Ein Name ist keine Eigenschaft eines Objekts, sondern ein anderes Objekt, welches dem benannten Objekt zum Zwecke der Identifikation zugeordnet wurde. Trotzdem wäre es in einer Datenbeschreibung unzuweckmäßig, Name und benanntes Objekt als zwei Objekte nebeneinanderzustellen. Intuitiv empfindet man einen Namen als etwas den Eigenschaften sehr ähnliches. Deshalb wird nun der Begriff des Attributes (= das Beigegebene) eingeführt,

welches der Strukturverdichtung dient und uns erlaubt, neben den Eigenschaften, die als Kontinuumswerte erlebt werden, auch Namen und andere diskrete Informationen als Inhalt der Objekte zu betrachten. Jedes Attribut ist also entweder eine Eigenschaft oder eine primär diskrete Information, die bei einer Strukturverdichtung dem Objekt zugeschlagen wurde. Die Wortkombination „primär diskret“ soll den Unterschied betonen zu den sekundär diskreten Eigenschaftswerten, die dadurch entstehen, daß das Kontinuum der Eigenschaftswerte durch Auflösungsbeschränkung in diskrete Intervalle aufgeteilt wird – beispielsweise bei Temperaturangaben in ganzen Graden.

#### 4 Aufbau von Datengittern

Eine Datenbeschreibung bezweckt entweder die Vermittlung eines bestimmten Datenwertes oder eines Wertebereichs, also beispielsweise entweder der natürlichen Zahl 5 oder des Wertebereichs INTEGER. Ein Datengitter zur Beschreibung eines bestimmten Wertes wird exemplarisches Datengitter genannt, ein Datengitter zur Beschreibung eines Wertebereichs soll allgemeines Datengitter heißen. Im Rahmen einer Software-Dokumentation interessieren primär die allgemeinen Datengitter; exemplarische Datengitter können aber zur plausiblen Hinführung zu einem allgemeinen Datengitter hilfreich sein. Datengitter sind nur dann sinnvoll, wenn der Weltausschnitt, der durch die zu beschreibenden Daten identifiziert werden soll, nicht als ein einziges Objekt erfaßt werden kann, sondern als eine Menge von Objekten gesehen werden muß, die Beziehungen zueinander haben.

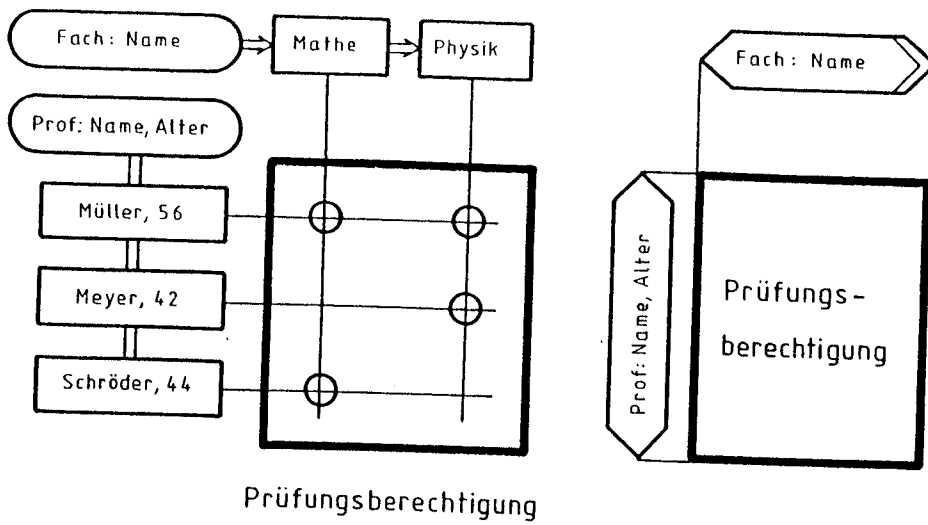
In einem Datengitter werden Objekte, Attribute und Beziehungen identifiziert, und zwar teilweise durch Charakterisierung und teilweise durch Benennung. Die in die Grafik eingetragene Beschriftung stellt eine Benennung dar, während die sichtbare Zuordnung zwischen der Beschriftung und den graphischen Elementen sowie die Beziehungen der graphischen Elemente untereinander als Charakterisierung anzusehen sind.

Die beiden Grundelemente zum Aufbau von Datengittern sind die Objektmenge und die Relationsmatrix. Anhand des Beispiels in Bild 4 werden diese Elemente vorgestellt: Im exemplarischen Datengitter kommen insgesamt fünf Objekte vor, die in die beiden Objektmengen „Professoren“ und „Fächer“ eingeteilt sind. Eine Objektmenge wird so dargestellt, als säßen die Objekte wie Perlen auf einer Schnur, wobei durch die vorhandenen oder fehlenden Pfeile ausgedrückt, ob die lineare Ordnung relevant oder irrelevant ist. Dies entspricht dem Unterschied, der durch unterschiedliche Klammern in der mathematischen Symbolik ausgedrückt wird: (a, b, c, d) bei relevanter Ordnung und {a, b, c, d} sonst.

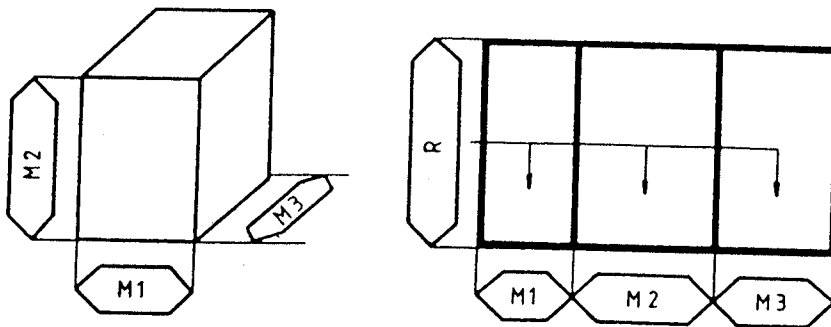
Die Erfassung von Beziehungen zwischen Objekten durch eine Matrix mit ausgewählten Kreuzungspunkten ergibt sich unmittelbar aus der mathematischen Definition einer Relation als Teilmenge eines kartesischen Produkts:

$$R \subseteq M_1 \times M_2 \times M_3 \times \dots \times M_n$$

Während das exemplarische Datengitter die einzelnen Objekte und Relationselemente zeigt, kann das allgemeine



**Bild 4**  
Ein exemplarisches Datengitter und seine Verallgemeinerung



**Bild 5**  
Darstellung von drei- und mehrstelligen Relationen

Datengitter nur noch die Existenz der einzelnen Objektmengen und Relationsmatrizen zeigen.

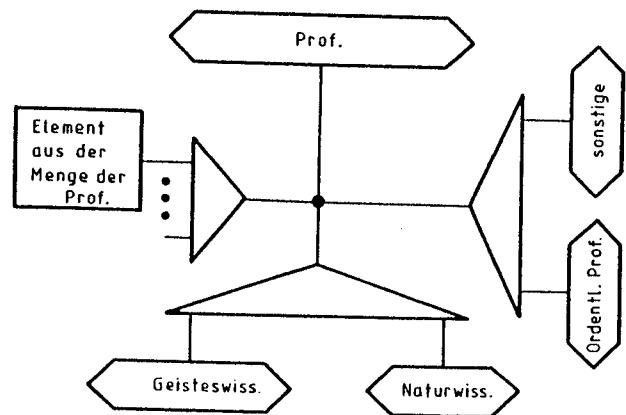
Da die Anzahl der Faktoren einer Relation größer als zwei sein kann, ist es manchmal erforderlich, die Menge der Relationselemente als Objektmenge darzustellen. Bild 5 zeigt dies für den Fall einer dreistelligen Relation: Die Elemente der Menge  $R$  sind ausgewählte Kreuzungspunkte des Quaders  $M_1 \times M_2 \times M_3$ , und deshalb muß jedes Element von  $R$  über eine entsprechende zweidimensionale Relationsmatrix eindeutig auf seine Koordinaten verweisen. Diese Relationsdarstellung durch Koordinatenverweis ist offensichtlich nicht auf Dreidimensionalität beschränkt.

Es muß möglich sein, nichtdisjunkte Teilmengen einer Objektmenge auch im allgemeinen Datengitter als grafische Elemente sichtbar zu machen, damit sie als Faktoren von Relationen auftreten können. Dazu wird das Partitionsymbol in Form eines flachen Dreiecks eingeführt. In Bild 6 wird die Menge der Professoren auf drei unterschiedliche Weisen partitioniert, wobei zweimal jeweils ein Binärkriterium die Aufteilung festlegt, während im dritten Fall die Aufteilung in Einzelelemente erfolgt, wobei aber nur ein einziges Element explizit als Rechteck stellvertretend für alle anderen dargestellt ist. In ein solches Objektrechteck kann wieder ein Datengitter eingezeichnet werden, wenn man einen strukturierten Aufbau zeigen will, der für alle Elemente der partitionierten Menge in gleicher Weise gilt. Ein Beispiel hierfür wird im Abschnitt 5 vorgestellt.

Bei der Einführung des Attributbegriffs wurde bereits darauf hingewiesen, daß die Attributierung ein Mittel zur

Strukturverdichtung ist. Dies soll nun anhand des Beispiels in Bild 7 veranschaulicht werden. Bild 7 zeigt zwei exemplarische Datengitter, die auf unterschiedliche Weise den gleichen Sachverhalt ausdrücken, d.h. die in unterschiedlicher Modellierung denselben Weltausschnitt identifizieren. Die eine Darstellung ist attributarm und relationsreich. Neben den vier Relationsmatrizen gibt es hier noch eine Relation zwischen Prüfungsteil und Prüfung, die nicht als Matrix, sondern als Partition dargestellt ist. Im allgemeinen Datengitter müßte diese fünfte Relation auch eine Matrix sein.

Das zweite Datengitter in Bild 7 ist im Vergleich zum ersten attributreich und relationsarm. Es gibt nur noch drei Objektmengen und zwei Relationen. Die Menge der Fächer und die damit verbundenen Relationen wurden durch entsprechende Attributierung der Professoren und



**Bild 6** Unterschiedliche Partitionierung einer Objektmenge

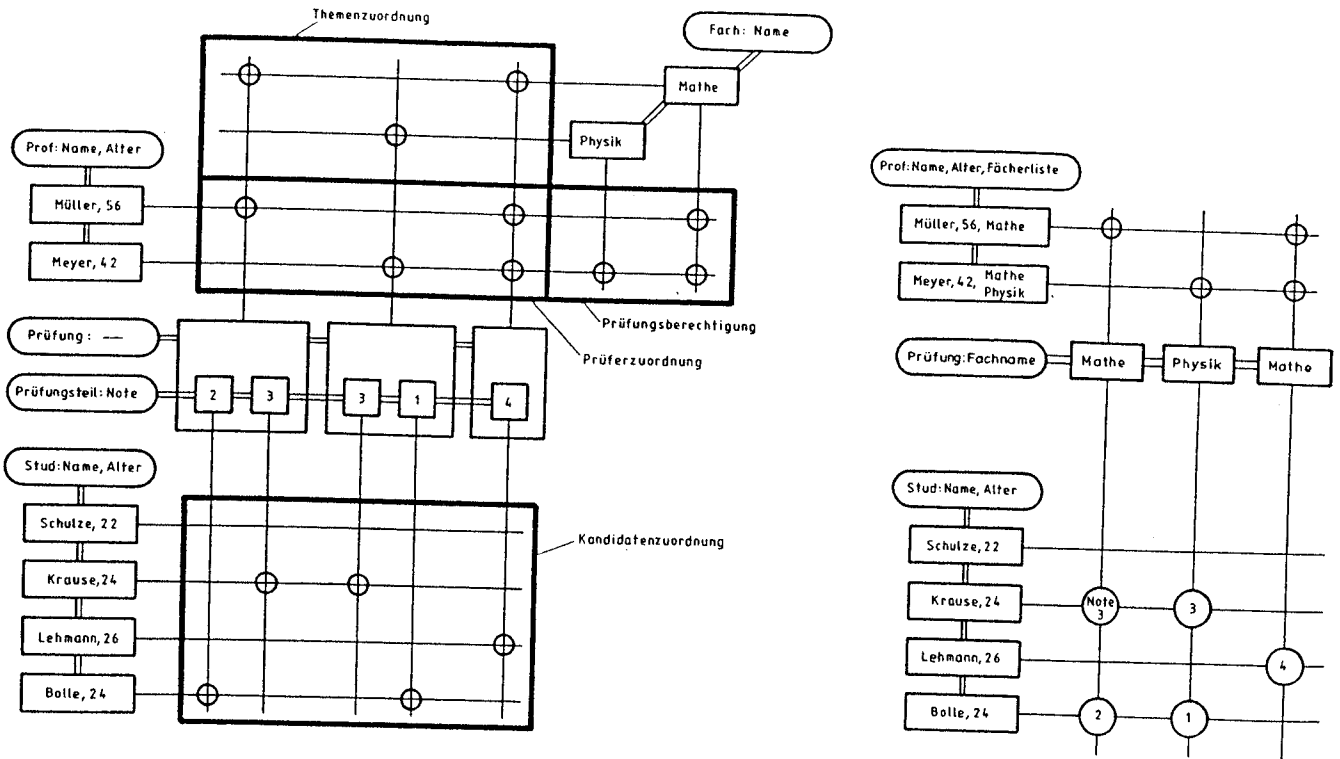


Bild 7 Strukturverdichtung durch Attributierung

der Prüfungen ersetzt. Die Menge der Prüfungsteile konnte entfallen, weil jedem Prüfungsteil umkehrbar eindeutig ein Element der Kandidatenzuordnung entspricht; die Elimination der Prüfungsteile als eigenständige Objekte erfordert aber nun eine Attributierung der Relationselemente in Form der Noten.

Zum Schluß dieses Abschnitts muß noch kurz etwas über die sogenannten Restriktionen gesagt werden. Eine Restriktion ist eine Bedingung, welche die zulässigen Mengmächtigkeiten, Attributierungen und Relationsstrukturen einschränkt. Eine Restriktion ist entweder logisch zwingend oder willkürlich. Am Beispiel in Bild 7 sei dies veranschaulicht. Eine logisch zwingende Restriktion lautet: Einer Prüfung darf nur ein Professor als Prüfer zugeordnet werden, wenn er die Prüfungsberechtigung für das zu prüfende Fach hat. Eine willkürliche Restriktion könnte lauten: Wenn es mindestens einen Professor unter 50 Jahren gibt, dürfen die Professoren, die 50 oder älter sind, keine Physik prüfen.

Die Frage, ob eine Restriktion eingehalten ist oder nicht, ist nur im Falle exemplarischer Datengitter sinnvoll; die Frage dagegen, ob eine Restriktion im Rahmen eines Datengitters sichtbar gemacht werden kann, kann sich nur auf allgemeine Datengitter beziehen. Die meisten Restriktionen lassen sich nicht anschaulich und übersichtlich im Datengitter darstellen. Dies ist aber kein wesentlicher Mangel, denn zur Vermittlung des Verständnisses der Semantik einer Datenstruktur leisten die Restriktionen i.a. keinen Beitrag. Deshalb verliert man nichts, wenn man die Restriktionen nur in einem Anhang zum Datengitter auflistet. Nur solche Restriktionen, die jeweils nur eine einzige Relation betreffen, lassen sich leicht im allgemeinen Datengitter symbolisieren, und zwar durch Eintragen entsprechender Symbole in die jeweiligen Relationsmatrizen. Das häufigste Symbol ist dabei der Ver-

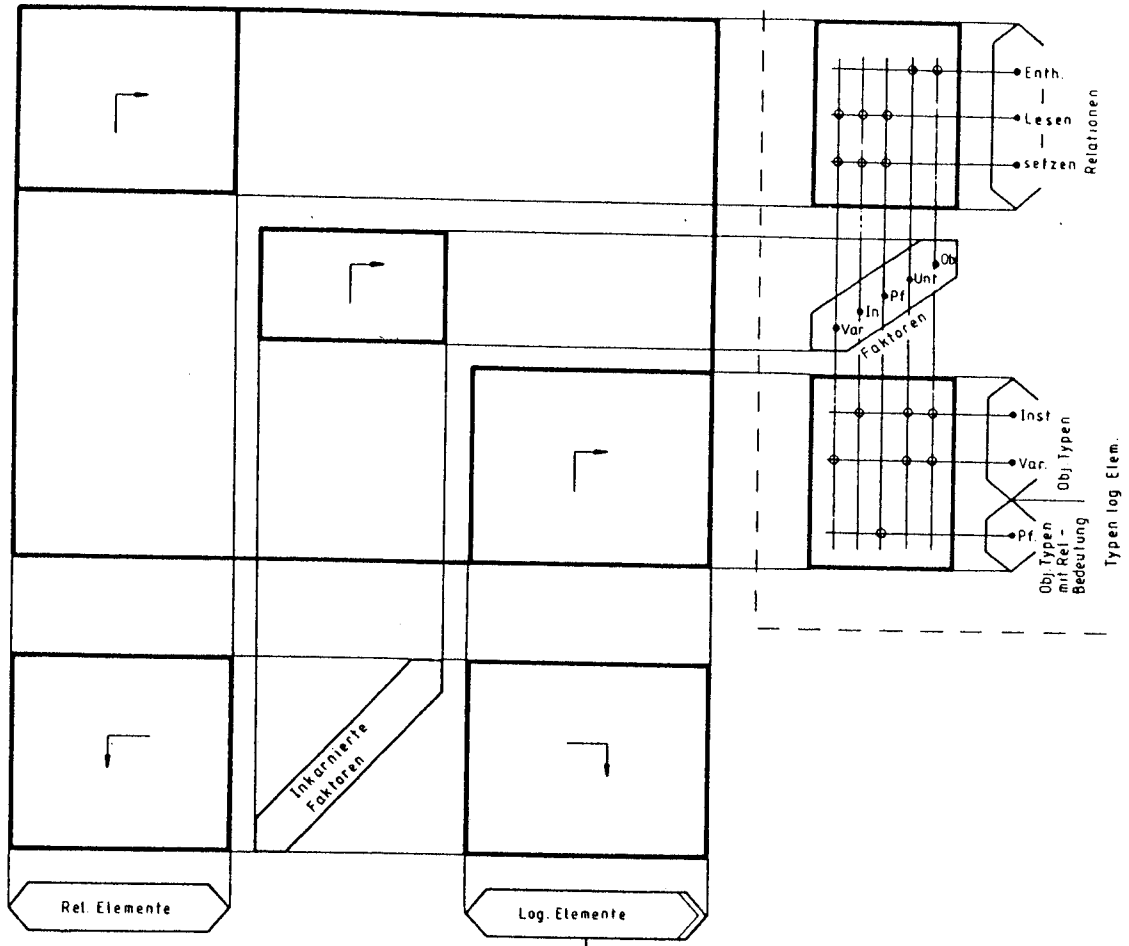
weisepfeil in eindeutigen Zuordnungen. Die Kandidatenzuordnung und die Themenzuordnung in Bild 7 sind von dieser Art; deshalb würde man im zugehörigen allgemeinen Datengitter in die entsprechenden Relationsmatrizen rechtwinklig abgeknickte Pfeile eingetragen, wie sie in Bild 8 vorkommen.

## 5 Datengitter zum Grafik-Editor

Bild 8 zeigt das Datengitter der Arbeitsdaten des Grafik-Editors in Bild 1. Durch gestrichelte Schnittlinien ist das Datengitter in drei Bereiche unterteilt, die getrennt kommentiert werden. Rechts oben ist das Datengitter der Plantypbeschreibung dargestellt. Weil hier das allgemeine Datengitter nur verständlich gemacht werden kann, indem es anhand eines exemplarischen Datengitters veranschaulicht wird, ist an dieser Stelle als exemplarisches Datengitter die Beschreibung des Plantyps „Instanzennetz“ eingetragen. Jeder Plan besteht aus graphischen Symbolen für logische Elemente. Im Falle der Instanzennetze gibt es drei Typen logischer Elemente: die Instanzen in Form von Rechtecken, die Speichervariablen in Form von Kreisen oder Ovalen, und die Zugriffspfade in Form von Pfeilen. Bild 9 zeigt ein Beispiel eines Instanzennetzes. Bei den logischen Elementen kann unterschieden werden zwischen ursprünglichen Objekten und objektifizierten Relationselementen. Im Falle der Instanzennetze gehören die Instanzen und Variablen in die eine Klasse und die Zugriffspfade in die andere, denn ein Zugriffspfad erhält seine Existenzberechtigung nur aus einer Beziehung zweier ursprünglicher Objekte.

Zwischen den logischen Elementen gibt es Relationen, wobei die Faktoren der karthesischen Produkte die Brücke bilden zwischen den Mengen unterschiedlicher Objekte.

Beschreibung des  
Plantyps "Instanzennetz"



Logische Struktur des Plans  
Grafische Struktur des Plans

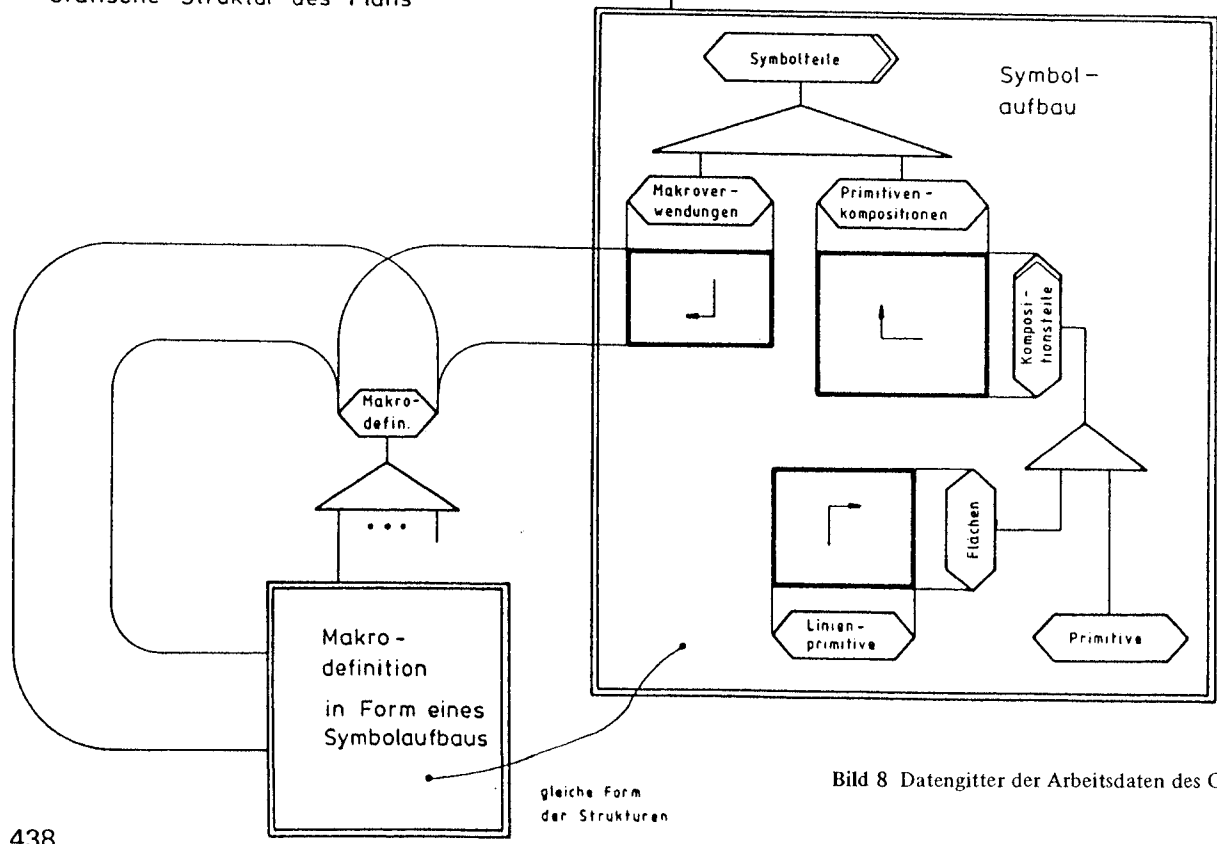


Bild 8 Datengitter der Arbeitsdaten des Grafik-Editors

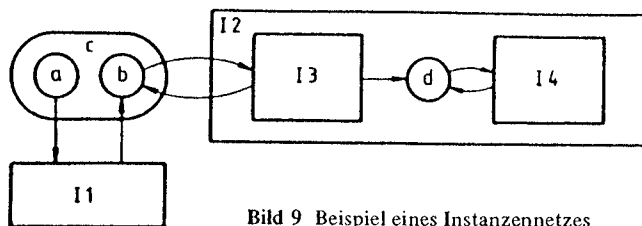


Bild 9 Beispiel eines Instanzennetzes

typen und den Relationsdefinitionen. Im Falle der Instanzennetze gibt es drei Relationen. Die Enthaltenseinsrelation ist zweistellig mit den beiden Faktoren „Oberelement“ und „Unterelement“, die beide gleich der Vereinigung der Instanzenmenge und der Variablenmenge sind. Die Restriktion, daß eine Variable keine Instanz enthalten darf, ist hier nicht sichtbar, hätte aber durch Einführung zweier unterschiedlicher Enthaltenseinsrelationen sichtbar gemacht werden können. Die beiden Zugriffsrelationen „Lesen“ und „Setzen“ sind jeweils dreistellig und haben jeweils die gleichen drei Faktoren, weil nämlich eine Beziehung zwischen Instanzen, Variablen und Zugriffspfaden ausgedrückt wird.

Der aktuelle Plan in Bearbeitung äußert sich als logische und als graphische Struktur. Zur logischen Struktur gehören drei Objektmengen, welche die Inkarnationen der in der Plantypbeschreibung eingeführten Typen logischer Elemente, Relationen und Faktoren sind. So ist beispielsweise der Plan in Bild 9 eine Inkarnation der Typs Instanzennetz, die aus folgenden drei Objektmengen besteht: (1) Menge der logischen Elemente mit 4 Instanzen, 4 Variablen und 7 Pfaden; (2) Menge der Relationselemente mit 3 Elementen des Lesens, 4 Elementen des Setzens und 5 Elementen des Enthaltenseins (Letztere sind: a in c, b in c, I3 in I2, d in I2, I4 in I2.); (3) Menge der inkarnierten Faktoren mit 31 Elementen, welche sich aus den Relationselementen und der jeweils zugehörigen Faktoranzahl ergeben ( $3 \cdot 3 + 4 \cdot 3 + 5 \cdot 2 = 31$ ).

Die drei Relationsmatrizen im Datengitter, welche die drei Mengen des aktuellen Plans mit den drei Mengen der Plantypbeschreibung in Beziehung bringen, sind in einem umfassenden Rechteck zusammengefaßt, um zum Ausdruck zu bringen, daß es sich hierbei um eine Inkarnationsrelation handelt. Das Kennzeichen einer solchen Relation besteht darin, daß nicht nur Objektmengen, sondern ganze Strukturen aus Objektmengen und Relationsmatrizen zueinander in Beziehung gebracht werden. So haben in Bild 8 nicht nur die drei Objektmengen der Plantypbeschreibung eine Entsprechung beim aktuellen Plan, sondern auch die beiden Relationsmatrizen der Plantypbeschreibung haben eine Entsprechung. Das gesamte allgemeine Datengitter der Plantypbeschreibung wird inkarniert durch Spiegelung an der Hauptdiagonalen der Inkarnationsmatrix.

Derjenige Teil des Datengitters, der die graphische Struktur eines Plans beschreibt, ist auf den Symbolaufbau eines logischen Elements beschränkt. Die graphischen Beziehungen zwischen den Symbolen verschiedener logischer Elemente sind nur durch Restriktionen unter Bezug auf die Attribute der graphischen Primitive zu formulieren und erscheinen deshalb im Datengitter nicht. So ist es beispielsweise selbstverständlich, daß sich das logische Enthaltensein auch

als graphisches Enthaltensein äußern muß, aber man kann einsehen, daß dies durch eine Strukturdarstellung in der Art des Datengitters nicht ausgedrückt werden kann.

Zu jedem logischen Element gehört ein Symbol, welches aus mehreren Teilen bestehen kann. Ein Symbolteil ist entweder eine Makroverwendung oder eine Primitivenkomposition. Die Bestandteile einer Primitivenkomposition sind Flächen und Primitive; letztere sind Strecken, Kreisbögen, Texte oder Punktmarkierungen. Die Flächen sind definiert über ihren Rand, der in Form von Linienprimitiven festgelegt wird.

Für die Mengen der logischen Elemente, der Symbolteile und der Kompositionsteile zeigt das Datengitter die Relevanz der linearen Ordnung. Dies ist begründet durch die Möglichkeit, daß sich graphische Komponenten überlappen und daß dann eine Überdeckungsreihenfolge festgelegt sein muß.

Die Restriktion, daß die Makroverwendung hierarchisch sein muß, d.h. daß sie nicht zyklisch sein darf, ist logisch zwingend, obwohl sie im Datengitter nicht zum Ausdruck kommt.

Es ist selbstverständlich, daß als nächstes die Attributierung des Datengitters vorgestellt werden müßte. Aber erstens würde dies den Umfang dieser Arbeit sprengen, und zweitens könnte dies zum Zweck des Beispiels, eine Vorstellung von der Brauchbarkeit der Datengitter zu vermitteln, nichts wesentliches mehr beitragen.

Zum Schluß muß unbedingt darauf hingewiesen werden, daß so ein Datengitter keinerlei Aussagen macht bezüglich der codierungstechnischen Gestaltung der Datenstruktur. Erst wenn man die Semantik der Daten verstanden hat, kann man darangehen zu überlegen, welche Teilinformationen man in welche Datensätze packt und wie man diese Sätze zu Feldern, Listen oder Netzen gruppiert. Hierbei spielt das Wissen um Häufigkeit und Art der Zugriffe zu den Teilinformationen eine große Rolle, was bei der Gestaltung des Datengitters völlig unerheblich ist.

#### Literatur

- [1] *Wendt, S.*: Projektbibliothek auf der Basis von Strukturplänen. In: *Angewandte Informatik* 11/84, S. 452–458
- [2] *Wirth, N.*: Algorithmen und Datenstrukturen. Teubner, Stuttgart, 1975
- [3] *Horowitz, E., Sahni, S.*: Fundamentals of Data Structures. Pitman, London, 1977
- [4] *Standish, T. A.*: Data Structure Techniques. Addison-Wesley, Reading, 1980
- [5] *Date, C. J.*: An Introduction to Database Systems. Addison-Wesley, Reading, 1977
- [6] *Wedekind, H.*: Datenbanksysteme I. Bibliographisches Institut, Mannheim, 1981
- [7] *Ortner, E.*: Semantische Modellierung – Datenbankentwurf auf der Ebene der Benutzer. In: *Informatik-Spektrum* 1/85, S. 20–28
- [8] *Chen, P. P.*: The Entity-Relationship-Model – Toward a Unified View of Data. In: *ACM Transactions on Database Systems* 1/1976, S. 9–36
- [9] *Rösner, W.*: Das Datengitter-Modell und seine Konsequenzen für den Entwurf informationsverarbeitender Systeme mit großen Datenbasen. Dissertation, Universität Kaiserslautern, 1983



# Angewandte Informatik

*Herausgeber:*

P. Schmitz, Universität zu Köln  
N. Szyperski, GMD, St. Augustin  
und Universität zu Köln

*Redaktion:*

Ulrich Hasenkamp, Köln  
Lehrstuhl für Informatik  
Robert-Koch-Str. 10  
5000 Köln 41  
Telefon (0221) 478 55 69

*Herausgeberrat:*

W. Ameling, Aachen  
H. Fiedler, Birlinghoven  
J. Griese, Bern  
E. Grochla, Köln  
R. Gunzenhäuser, Stuttgart  
Ch. Heinrich, Dortmund  
L. J. Heinrich, Linz/Österreich  
E. Henze, Braunschweig  
R. Jünemann, Dortmund  
W. Kämmerer, Jena/DDR  
G. Krüger, Karlsruhe  
H. Maurer, Graz/Österreich  
H. G. Pärli, Dortmund  
P. J. Pahl, Berlin  
P. L. Reichertz, Hannover  
B. Schmidt, Erlangen  
D. Seibt, Essen

---

**Bestellschein**

Verlag Vieweg, Postfach 5829, D-6200 Wiesbaden 1

<input type="checkbox"/> 1 Jahr DM 252,- (1986) zuzüglich Versandkosten Jahrgang 28, 1986, 12 Hefte jährlich
<input type="checkbox"/> 2 Jahre DM 454,- (1986/87) zuzüglich Versandkosten Jahrgang 28 + 29
<input type="checkbox"/> Probeheft

**Angewandte Informatik  
applied informatics**

Name	
Adresse	
Datum	Unterschrift