

**Heft 82**

**Carsten Berkau**

**VOKAL**

**(System zur Vogangskettendarstellung  
und -analyse**

**Struktur der Modellierungsmethode -**

**Juni 1991**

**C. Berkau:**

**VOKAL (System zur Vorgangskettendarstellung  
und -analyse)**

**- Struktur der Modellierungsmethode -**

Juni 1991

## **VOKAL (System zur Vorgangskettendarstellung und -analyse)**

### **- Struktur der Modellierungsmethode -**

#### **Inhalt:**

#### **0 Einleitung**

#### **1 Anforderungen an die Methode zur Darstellung und Analyse von Vorgangsketten**

#### **2 Ziele der Methode zur Darstellung und Analyse von Vorgangsketten**

#### **3 Methodenbeschreibung**

##### **3.1 VKD-Struktur**

##### **3.2 Metamodell des VKD**

##### **3.3 Elementdefinition und Attributierung der VKD-Elemente**

##### **3.4 Darstellung und Einteilung der verwendeten Symbole**

##### **3.5 Symbolverknüpfungsregeln**

#### **Literatur**

## 0 Einleitung

Eine Optimierung der Informationsverarbeitung im Unternehmen kann nur durch das Zusammenwirken der in organisatorischen Bereichen verteilten Informationssysteme erfolgen. Zur Gewährleistung einer Kooperation von Anwendungssystemen und einer redundanzfreien Speicherung von Informationen leistet die Informationsmodellierung insbesondere in der Planungsphase von Informationssystemen einen substantiellen Beitrag. Die Anzahl von auf dem Softwaremarkt angebotenen CASE-Werkzeugen zur Unterstützung der Applikationssoftwareentwicklung unterstreicht die aufgezeigte Entwicklung. Neben vielen Unternehmen, die sich aktuell mit der Entwicklung von Daten- und Funktionsmodellen zur Optimierung ihrer Applikationen beschäftigen, veröffentlichen zunehmend auch die Anbieter von Standardsoftware die ihren Systeme zugrundeliegenden Fachkonzeptionen. Im Vorfeld der fachkonzeptionellen Modellierung von Informationssystemen steht die Erfassung und Analyse von betrieblichen Vorgangsketten. Durch die Analyse von Vorgangsketten - z.B. mit Vorgangskettendiagrammen - werden Schwachpunkte innerhalb der Ablaufstruktur von Vorgangsketten deutlich und geben Anhaltspunkte zur Ablaufoptimierung. Die im folgenden dargestellte Methode soll die Erfassung und Analyse von VKD unterstützen und dient als konzeptionelle Grundlage eines Tools zur Vorgangskettenerfassung und Analyse VOKAL.

Weil das System VOKAL neben der Abbildung und Auswertung von Vorgangsketten auch die Basis einer fachkonzeptionellen Informationsmodellierung darstellen soll, wird zunächst eine Einordnung des Systems in einen Modellierungsrahmen vorgenommen. Vertreter aus Forschung und Praxis beschäftigen sich derzeit intensiv mit der Entwicklung einer Modellierungsarchitektur, in die einzelne zur Komplexitätsreduktion getrennte Modellierungssichten eingeordnet werden können. Weil die Architektur integrierter Informationssysteme (ARIS) (Vgl. Scheer, A.-W.: 1991.) neben der Betrachtung des Modellierungsgegenstands aus verschiedenen Sichten zusätzlich das Zusammenwirken der einzelnen Sichtweisen untereinander berücksichtigt, soll die ARIS-Architektur den weiteren Ausführungen zugrunde liegen. Die Entwicklung der Methode zur Vorgangskettendarstellung und -analyse ist jedoch nicht unmittelbar von der gewählten Architektur abhängig.

Betriebliche Vorgänge als Elemente einer Vorgangskette lassen sich durch ein Vorgangskettenmodell darstellen (Vgl.: Scheer, A.-W.: 1991, S. 13ff.). Dieses stellt den Bezug zwischen Vorgängen und den sie beeinflussenden Faktoren dar. Weil die vielfältigen Einflußfaktoren auf einen Vorgang eine komplexe und redundante Vorgangsbeschreibung - insbesondere bei der Modellierung von Vorgangsfolgen - bewirken, teilt man die Modellierung in einzelne Sichten ein und betrachtet dabei nur die jeweils sichtenspezifischen Faktoren. Das Bild 1 zeigt das Vorgangskettenmodell, bei dem die Faktoren, die den Vorgang beeinflussen, einzelnen Sichten zugewiesen sind.

In der Ablaufsteuerungssicht wird der Zusammenhang zwischen den einzelnen Sichten, der bei der Aufteilung in einzelnen Sichten verloren geht, dargestellt.

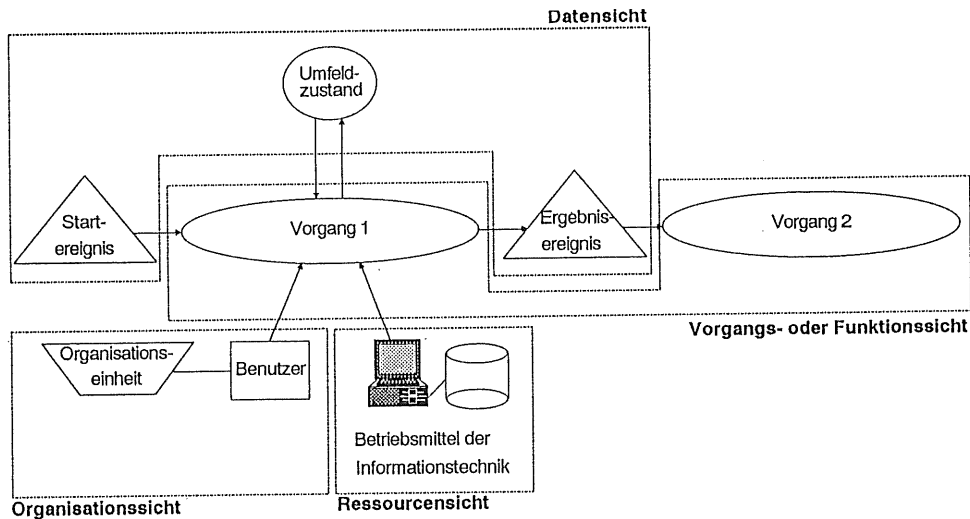


Bild 1: Sichten des Vorgangskettenmodells (aus: Scheer, A.-W.: 1991, S. 14)

Die an Sichten wie Daten-, Funktions-, Organisations- und Ablaufsteuerungssicht orientierte Unternehmensmodellierung erfolgt auf den Ebenen des der Softwareentwicklung zugrundeliegenden Phasenmodells. Es werden die Phasen Fachkonzept, EDV-Konzept und Implementierung differenziert (Vgl.: Scheer, A.-W.: 1991). Die Phasen laufen nicht entsprechend eines Wasserfallmodells streng nacheinander ab, sondern erlauben Schnelldurchläufe (Prototyping) oder lassen Rücksprünge zwischen den Phasen zu.

Das VKD, das die methodische Basis des Systems VOKAL darstellt, wird in die ARIS-Architektur als Beschreibungssprache der realen Abläufe eingeordnet und stellt die Informationen für eine fachkonzeptionelle Modellierung bereit. Durch das Herstellen des Bezugs zu den sichtenspezifischen Fachmodellen kann das VKD auch auf der Ebene des Fachkonzepts in der Ablaufsteuerungssicht eingeordnet werden. Das VKD stellt eine Abfolge von Vorgängen sichtenübergreifend dar. Wie das Bild 2 verdeutlicht, wird ein Vorgang aus Organisationssicht (Abteilung), Funktionssicht (Vorgang) und der Verbindung von Funktion- und Datensicht (Bearbeitung, Datenbasis) modelliert.

Das Bild 2 zeigt exemplarisch ein VKD im Bereich der Auftragsbearbeitung.

VKD erlauben eine Darstellung von Vorgangsketten entsprechend des Ablaufs und somit quer zur am Organisationsdiagramm orientierten Funktionsbereichseinteilung. Dadurch erlauben sie eine Modellierung von ganzheitlichen Vorgangsketten, die frei von dem Tayloristischen Grundsatz der Arbeitsteilung und von Bereichseinflüssen sind. Vorgänge werden mit VKD nicht länger bereichsorientiert betrachtet und optimiert,

sondern können entsprechend der Philosophie des Computer Integrated Manufacturing (CIM) durch eine am Gesamtprozeß orientierte Denkweise analysiert und gestaltet werden.

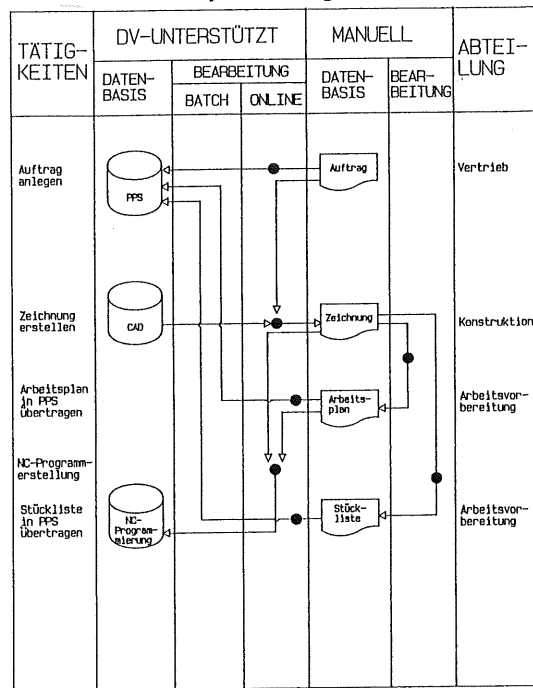


Bild 2: VKD Auftragsbearbeitung

Das VKD eignet sich zur implementierungsnahen Abbildung von sequentiellen Vorgangsketten. Bei der Abbildung der Vorgänge werden neben den funktionalen Betrachtungen auch die Organisations-, die Daten- und die Ressourcensicht mit einbezogen. Die Abläufe werden entsprechend den in der Realisierung vorhandenen Objekten modelliert, so daß VKD aufgrund ihres geringen Abstraktionsgrads leicht verständlich sind. Sie eignen sich sowohl als Beschreibungssprache für Vorgänge innerhalb der Fachabteilungen bei der Erfassung und Verifizieren von Vorgangsketten als auch im Rahmen von Präsentationen auf der Managementebene.

VKD unterscheiden sich von den Fachmodellen zum einen durch einen geringeren Abstraktionsgrad, zum anderen durch die Tatsache, daß keine Aufteilung der Vorgangskettendarstellung in Sichten (Daten, Funktionen, Organisation, Ressourcen) erfolgt. Im VKD kann eine sichtenübergreifende Abbildung von Vorgangsketten realisiert werden, weil Abläufe von Vorgängen als sequentielle Abfolge dargestellt werden.

Die von Scheer entwickelten VKD wurden bislang im Rahmen von Industrieprojekten erfolgreich bei der Beratung bezüglich unternehmensweiter ablauforganisatorischer Fragestellungen eingesetzt und wurden entsprechend den projektspezifischen Anforderungen spezialisiert und modifiziert. Die Projekterfahrung mit VKD sowie deren Weiterentwicklungen und Varianten fließen inhaltlich in die Konzeption des Systems VOKAL ein.

Die Erstellung und Analyse von Vorgangskettendiagrammen wurde bisher nicht DV-technisch unterstützt. Computerunterstützung besteht derzeit lediglich im Rahmen der zeichentechnischen Umsetzung von Vorgangskettendiagrammen. Aufgrund des hohen Analysepotentials, das durch eine Computerunterstützung effizienter ausgeschöpft werden kann, sowie wegen der leichteren Handhabbarkeit der Methode zur Vorgangskettenerstellung hat man die Notwendigkeit einer DV-Unterstützung der skizzierten Methode erkannt. Es soll deshalb ein DV-System zur Erfassung und Analyse von VKD entwickelt werden.

Im folgenden wird hierzu in einem ersten Schritt die Methode zur Darstellung und Analyse von VKD definiert und dient als Grundlage für die Fachkonzeptionierung des zu entwickelnden Computersystems VOKAL.

VOKAL soll als PC-System in der Lage sein, die Erfassung funktionale Abläufe im Unternehmen als Vorgangskettendarstellung zu unterstützen und im Rahmen von ablauforganisatorischen Analysen betrieblicher Vorgangsketten zur Identifikation von Rationalisierungspotentialen beitragen zu können.

Die VKD-gestützte Darstellung und Analyse betrieblicher Funktionsabläufe zur Unterstützung von unternehmensweiten Rationalisierungsprojekten erfolgt bislang entsprechend den im folgenden dargestellten Phasen. Dabei ist für die Betrachtung irrelevant, ob und in welchem Umfang die Methode durch DV-Anwendungen unterstützt wird.

- Erfassung und Darstellung des Istzustandes;
- Analyse des Istzustandes;
- Identifikation von Verbesserungspotentialen;
- Erstellung und Bewertung eines Sollkonzeptes;
- Realisierung ablauforganisatorischer Modifikationsvorschläge.

Betriebliche Vorgangsketten werden zunächst mit Hilfe von Interviewtechniken in Zusammenarbeit mit den Experten die den Fachabteilungen erhoben und zeichentechnisch in handskizzierte VKD-Darstellungen umgesetzt. Anschließend werden die VKD mit den Interviewpartnern verifiziert und gegebenenfalls modifiziert, bis die dargestellten Abläufe mit den realen kongruent sind. Im Rahmen der Analyse des Istzustandes werden die Abläufe untersucht und bewertet. Kennzahlen für Eigenschaften betrieblicher Abläufe ermöglichen dabei eine Quantifizierung von Vorgangskettenbewertungen. Mögliche Kennzahlen können beispielsweise die Anzahl von Abteilungswechseln, die Anzahl der ausführbaren Funktionen pro Vorgangskette, etc. sein. Die

aufgenommenen VKD werden anschließend auf ablauforganisatorische Schwachstellen und mögliche Rationalisierungspotentiale untersucht. Die ermittelten Schwachpunkte der analysierten Vorgangskette stellen die Grundlage der Ableitung von Modifizierungsmaßnahmen für die Ablauforganisation betrieblicher Vorgänge dar. Die Vorschläge für die Umgestaltung der Vorgangskette werden in einem Sollkonzept, das ebenfalls durch ein VKD visualisiert und bewertet werden kann, abgebildet. Dieses VKD erfolgt auf einem höheren Detaillierungs- und Konkretisierungsgrad als das VKD zur Erfassung der Istsituation. Nach der Bewertung des Konzeptes und einem Soll-Ist-Vergleich für die jeweiligen Vorgangsketten kann deren ablauforganisatorische Umgestaltung mit Hilfe von Kosten-Nutzen-Analyseverfahren bewertet werden und bietet somit die Grundlage für die Entscheidungsfindung bezüglich Umstrukturierungsinvestitionen der Ablauforganisation innerhalb der betrachteten Vorgangskette. Darüber hinaus kann das VKD als Basis zur Entwicklung von sichtenspezifischen Fachmodellen angesehen werden.

Im folgenden werden die Anforderungen und Ziele der Methode zur Darstellung und Analyse von Vorgangsketten mit VKD beschrieben und die Elemente der Methodik anhand eines Meta-Modells in der Beschreibungssprache des Entity-Relationship-Modells dargestellt und erläutert.



## **1 Anforderungen an die Methode zur Darstellung und Analyse von Vorgangsketten**

Die dargestellte Methode zur Darstellung und Analyse von VKD muß explizit den im folgenden aufgeführten Anforderungen gerecht werden:

### **Methodische Eindeutigkeit**

Eine Vorgangskette muß sich mit einem VKD eindeutig abbilden lassen. Es muß ausgeschlossen werden, daß relevante Aspekte der Vorgangsketten unscharf oder mehrdeutig darstellbar sind.

### **Teamfähigkeit**

Teamfähigkeit wird gefordert, wenn die Methode von einem Team angewendet werden soll. In diesem Fall muß gewährleistet werden, daß alle Mitglieder des Teams zueinander passende und konsistente Ergebnisse liefern.

### **Einfache Verständlichkeit**

Die Methode soll einfach verständlich und anschaulich sein. Die leichte Verständlichkeit stellt eine wesentliche Voraussetzung zur Akzeptanz der Methode dar. Die Anwender der Methode sollten diese ohne umfangreiche Schulungsmaßnahmen in Kürze erlernen und anwenden können. Die Methode wird damit fehlerunanfällig. Dieses wird dadurch unterstützt, daß die Vorgangskettendarstellung am realen Dokument und Belegfluß ausgerichtet ist. Es werden demzufolge keine abstrakten Modelle aufgebaut, sondern realitätsnahe Objekte in den VKD abgebildet. So kann gewährleistet werden, daß die Methode im Rahmen von Experteninterviews und Präsentationen aufgrund ihrer eingängigen, weil realitätsbezogenen, Darstellung verwendbar ist.

Zur Transparenzerhöhung sollte darüber hinaus die Anzahl der Konstrukte einer Beschreibungsmethode minimiert werden. Prinzipiell ist eine möglichst generische Konstruktdefinition zu fordern, so daß die im ersten Schritt per Interview erhobenen Vorgangsketten näher spezifiziert werden können. Beispielsweise soll es vermieden werden, eine Fülle spezieller Symbole für ähnliche Inhalte zu entwickeln. Vielmehr sind dem Modellierer allgemeingültige Symbole anzubieten, die durch Zusätze spezifizierbar sind. Somit wird ausgeschlossen, daß die Modellierung von Vorgangsketten vor Ort durch Zuordnungsprobleme zwischen realen Objekten und im VKD darzustellenden Symbolen behindert wird. Es sollte eine einfache und eindeutige Zuordnung realer Objekte zu den Darstellungssymbolen angestrebt werden und die Option einer anschließenden Spezifizierung offengehalten werden. Beispielsweise kann der Datenspeicher für einen Auftrag

im VKD zunächst als ein allgemeines Papierdokument (man. Unterlage) beschrieben werden und bei der Nachbereitung der modellierten VKD als ein Auftragsformular spezifiziert werden. Dem Auftrag kann dann ebenso ein Zustand als beschreibendes Attribut zugeordnet werden.

Ein weiterer Vorteil einer geringen Anzahl von Konstruktionselemente einer Methode kann in dem entsprechenden Aufwand zur Definition von Regeln zur Anwendung der Methode gesehen werden. Die Konstrukanzahl steht in direkt proportionalem Verhältnis zu der Anzahl der Regeln zur Verknüpfung der Konstrukte.

### **Überprüfbarkeit**

Regeln einer Methode sollten so gestaltet werden, daß sie leicht und möglichst formal überprüfbar sind. Dieses bietet die Möglichkeit, die korrekte Anwendung der Methode schnell und unkompliziert überprüfen zu können. Speziell bei einer Methode, die DV-technisch umsetzbar sein soll, müssen sich Regeln formal überprüfen lassen. In dem Fall besteht die Möglichkeit, das Einhalten von Regeln durch Plausibilitätsprüfungen systemunterstützt abzurufen. Bei umfangreichen Modellierungsprojekten von Vorgangsketten ist hierin ein großer Vorteil systemgestützter Abbildungs- und Analyseapplikationen zu sehen. Das Vorsehen von Plausibilitätsprüfungen zur Eingabekontrolle kann dabei methodisch unkorrekte Eingaben sofort unterbinden und erspart nachfolgende Korrekturmaßnahmen an Vorgangsketten.

### **Übersichtlichkeit der Darstellung**

Die Zielsetzung einer VKD-Darstellung ist eine leicht überschaubare Darstellung von Vorgangsketten zu erzeugen, die gleichzeitig Ausgangspunkt für die Analyse der dargestellten Vorgangsketten sein kann. Ablauforganisatorische Schwachpunkte der abgebildeten Funktionsfolge sollten frühzeitig aufgrund der Darstellung offensichtlich sein. Demzufolge muß die Abbildung der Vorgangsketten mit Hilfe von VKD möglichst übersichtlich, anschaulich und strukturiert sein.

### **Änderungsfreundlichkeit**

Insbesondere bei der Entwicklung einer Beschreibungsmethode, die auch technische Vorgänge abbilden kann, ist darauf zu achten, daß die Methode änderungsfreundlich zu gestaltet ist. Treten Änderungen des dargestellten Inhalts auf, müssen diese direkt in der Darstellung nachvollzogen werden können, bzw. muß das entwickelte VKD leicht modifizierbar sein. Eine zu fordernde Versionsverwaltung des zu entwickelnden Systems steht mit der Änderungsfreundlichkeit direkt in Bezug. Diese Forderungen sind an die Funktionalität des Systems zu stellen und müssen bereits im Rahmen der Methodendefinition berücksichtigt werden. Dieses

um so mehr, weil die Erfassung betrieblicher Vorgangsketten per Interviewtechnik nicht in jedem Fall sofort die Abläufe in realitätsgetreuer Form wiederzugeben in der Lage ist. Bei Reviews mit den Experten werden deshalb die Abläufe oftmals korrigiert, so daß für eine computerunterstützte Modellierung die Änderungsfreundlichkeit des Systems (d.h. auch der Modellierungsmethodik) ein wesentliches Beurteilungskriterium darstellt.

### **Anpaßbarkeit bzgl. technischer Änderungen**

Methoden zur Darstellung realer Vorgangsketten müssen Veränderungen von Vorgängen folgen können. Treten bei betrieblichen Funktionsabläufen technisch bedingte Veränderungen auf, sollte die Methode zur Abbildung der veränderten Sachverhalte erweiterbar gestaltet werden. Erfolgt beispielsweise die Datenspeicherung zukünftig auf neuartigen Speichermedien, muß die Methode um die Darstellung der jeweiligen verwendeten Speicherungsmöglichkeit erweitert werden. Im Fall der zu entwickelnden Methode zur Vorgangskettendarstellung und -analyse muß der Symbolvorrat beispielsweise durch einen Symbolgenerator erweiterbar gestaltet werden. Die Erweiterbarkeit bezieht sich auf alle Methodenkonstruktionselemente.

### **Freiheitsgrade der Darstellung**

Vorgangsketten sollen mit der Methode realitätsgetreu abbildbar sein. Dabei darf die Darstellung von VKD nicht durch unnötige Restriktionen eingeschränkt werden.

### **Verwaltung mehrerer Projekte**

Mit dem System VOKAL sollen mehrere VKD-Projekte bearbeitet werden können. Zur vorgangskettenorientierten Analyse ist es deshalb erforderlich, den Bezug der VKD-Elemente zu einer Vorgangskette zu gewährleisten. Beispielsweise wird aus diesem Grund auch keine redundanzfreie Speicherung von Vorgängen, die innerhalb verschiedener Vorgangsketten enthalten sind, verfolgt. Änderungen eines Vorgangs könnten in dem Fall die Modifikation anderer VKD bewirken.

## 2 Ziele der Methode zur Darstellung und Analyse von Vorgangsketten

Die Methodenziele sind von den Zielen und Funktionalitätsansprüchen des zu entwickelnden Systems zur Vorgangskettenerfassung und -analyse zu trennen. Es werden an dieser Stelle lediglich die an die Methode zu stellenden Ziele dargestellt:

- Visualisierung von realen unternehmensweiten Vorgangsketten unter Berücksichtigung von Aspekten aus den Sichten:
  - Funktionen,
  - Daten,
  - DV-Ressourcen,
  - Organisation;
- Abbildbarkeit von Zusammenhängen zwischen den Sichten
- Veranschaulichung von ablauforganisatorischen Merkmalen der Vorgangsketten wie:
  - Wechsel von DV-Systemen,
  - Medienbrüche ( z.B. von Papier zu Datenträger),
  - Wechsel von organisatorischen Einheiten,
  - Dokumentfluß;
- Visualisierung von allen vorgangskettenrelevanten Merkmalen;
- Abbildung von möglichen alternativen Teilvorgangsketten;
- Darstellung von Sollabläufen betrieblicher Vorgangsketten;
- implementierungsgetreue Abbildung von Vorgangsketten;
- Darstellungsmöglichkeit von Verfeinerungen von Vorgängen
- Integration der VKD in den CIM-Analyzer (Vgl.: Jost, W. et al.: 1991).

### 3 Methodenbeschreibung

Die Beschreibung der Methode zur Darstellung und Analyse von Vorgangsketten erfolgt in den Schritten:

- 3.1 VKD-Struktur;
- 3.2 Metamodell des VKD;
- 3.3 Elementdefinition und Attributierung der VKD-Elemente;
- 3.4 Darstellung und Einteilung der verwendeten Symbole;
- 3.5 Symbolverknüpfung.

Die Methode zur Vorgangskettendarstellung und -analyse basiert auf den von Scheer entwickelten VKD (Vgl. Scheer, 1990a, S. 7ff). Sie erlauben eine sequentielle Darstellung betrieblichen Funktionsablaufstrukturen in Diagrammform. Die Struktur dieser Diagramme, die die zu beschreibende Methode direkt beeinflusst, wird im folgenden beschrieben.

#### 3.1 VKD-Struktur

Die Vorgangskettenstruktur ist spaltenorientiert. Durch die Spaltenorientierung erlauben die VKD eine sichtenübergreifende Darstellung von Vorgangsketten. Im VKD werden betriebliche Funktionsabläufe im Gesamtkontext dargestellt. Damit wird gewährleistet, daß die Funktionalität von Vorgangsketten über aufbauorganisatorische Grenzen innerhalb des Unternehmens ebenso wie über Unternehmensgrenzen hinweg modelliert werden kann. Dieses entspricht der Philosophie von CIM, Abläufe im Unternehmen bereichsunabhängig zu betrachten. Wesentlicher Vorteil dieser Vorgehensweise kann in der integrativen Bearbeitung von Vorgangsketten gesehen werden. Diesem Aspekt muß auch die Modellierungsmethode Rechnung tragen. Nur so kann gewährleistet werden, daß die zu einer Vorgangskette gehörigen Vorgänge aufeinander aufbauend und entsprechend aufeinander abgestimmt modelliert und umgesetzt werden.

Es ist bei der Darstellung von betrieblichen Vorgangsketten zum Zweck der Planung der Ablaufoptimierung zu fordern, daß ablaufbezogene Betrachtungen von Funktionen sich auf den Gesamtablauf beziehen und somit auch die Vorgänge berücksichtigen, die in verschiedenen aufbauorganisatorisch zu trennenden Abteilungen eines Unternehmens ausgeführt werden. Deshalb werden bei Vorgangskettenbetrachtungen mit Hilfe der VKD immer ganzheitliche und abgeschlossene Vorgangsketten modelliert. Eine Vorgangskette gilt dann als abgeschlossen, wenn aus dem letzten Vorgang ein Ergebnis resultiert, das keinen weiteren Vorgang anstößt.

Das VKD im System VOKAL beinhaltet für die Abbildung von ganzheitlichen Vorgangsketten die drei Darstellungsfelder:

Vorgangskettendarstellung darstellen, symbolisch als Folge von über Informationsfluß- und Kontrollflußbeziehungen miteinander verknüpften Vorgangssymbolen inklusive den zugehörigen Datenspeicher- und Ereignissymbolen abgebildet. Das in der Mitte angeordnete Feld Vorgangskettendarstellung zur Visualisierung der Ablaufstruktur ist zur Strukturierung in einzelne Spalten unterteilt, so daß die Vorgänge und Datenträger einer Vorgangskette entsprechend den gebildeten Einteilungen klassifiziert abgebildet werden können. Die dabei verwendeten Symbole sind an die standardisierte Darstellungsform entsprechend DIN 66001 angelehnt, entsprechen diesen aber aufgrund der definierten Anforderungen an die Methode (s.o.) nicht exakt. Die rechts neben der Vorgangskettendarstellung angeordnete Spalte beinhaltet die Ressourcen. Ressourcen sind Personen, organisatorische Einheiten oder Rechnersysteme. Darüber hinaus werden in der Spalte Ressourcen Schnittstellen zu anderen VKD (Konnektoren) und zu organisatorischen Bereichen (Verteilung) modelliert.

Im Vergleich zur ursprünglichen VKD-Struktur ergeben sich folgende strukturelle Veränderungen des VOKAL-VKD:

### **Einführung der Spalte Ereignis**

In der Spalte Ereignis werden die Vorgänge auslösenden Ereignisse und aus Vorgängen resultierenden Ereignisse modelliert. Sie werden über Kontrollflußbeziehungen mit den Vorgangsdarstellungssymbolen verbunden. Die Darstellung von Ereignissen und Kontrollflüssen erfolgt optional.

### **Umbenennung der Spalte Abteilung in Ressourcen**

In der Spalte Ressourcen werden die Aufgabenträger abgebildet. Aufgabenträger können organisatorische Einheiten, Personen oder Computersysteme sein. Weil sich die Aufgabenträger unter dem Begriff Ressource subsumieren lassen, wird die Spalte in Ressourcen umbenannt.

### **Veränderung der Spaltenanordnung bei der Vorgangskettendarstellung**

Aufgrund der Integration des Kontrollflusses in das VOKAL-VKD ist eine Umstrukturierung der Spaltenfolge bei der Vorgangskettendarstellung erforderlich. Ziel bei der Spaltenanordnung war es, möglichst geringe Vermischung zwischen Informationsfluß- und Kontrollflußdarstellungen zu erreichen. Deshalb wurde die Spalte Ereignisse rechts von der Bearbeitungsspalte und die Datenbasis-Spalten links davon angeordnet. Im Gegensatz zum ursprünglichen VKD erfolgt die Modellierung von Vorgängen in einer gemeinsamen Spalte. So wird keine kategorische Trennung in DV-unterstützt und manuell erzwungen. Bei der Modellierung kann der Anwender bestimmen, in welchem Umfang der Vorgang DV-unterstützt bzw. manuell ausgeführt wird. Vollständig DV-unterstützte Vorgänge werden ganz links in der Bearbeitungsspalte, manuelle Vorgänge ganz rechts modelliert. Bei der Realisierung des Systems VOKAL ist für die Spalte Bearbeitung eine Rasterung vorgegeben. Nach ihrem DV-Unterstützungsanteils werden die Vorgänge in der Spalte Bearbeitung entsprechend der eingestellten

- Vorgänge;
- Vorgangskettendarstellung;
- Ressource.

Die Felder sind nebeneinander angeordnet und stehen durch Symbole verbindende Flußdarstellungen, die die VKD-Felder überschreiten, miteinander in Beziehung. Damit ist eine sichtenübergreifende Abbildung von Vorgangsketten gewährleistet. Weil die Vorgänge sequentiell modelliert werden, werden Vorgänge, die nacheinander ablaufen, untereinander dargestellt. Eine zeitliche Betrachtung der Vorgangskette im Sinne eines Vorgangsketten-Zeit-Diagramms mit einer zeitlichen Koordinate in vertikaler Richtung ist nicht das Ziel eines Vorgangskettendiagramms. Bei der Methode MERISE wird diese Art der Vorgangsmodellierung verfolgt. Es können dann aber nur rein sequentielle Vorgangsketten modelliert werden, Verzweigungen oder schleifenähnliche Konstrukte sind nicht darstellbar.

Das Bild 3 zeigt die Struktur des VOKAL-VKD. Neben den zur Vorgangskettenmodellierung vorgesehenen Feldern enthält jedes VKD einen Diagrammkopf, in dem Informationen über die dargestellte Vorgangskette, die Erstellung sowie ordnungsspezifische Informationen zum VKD eingetragen werden können.

Projekt:		Bearb.:		Vers.:		Datum:		Seite:	
Vorgang	Datenbasis		Bearbeitung		Ereignis	Ressource			
	DV-unterstützt	manuell	DV-unterstützt	manuell					

Bild 3: Struktur des VOKAL-VKD

Von den im VKD modellierten Vorgängen sind die Funktionen abzugrenzen, die als Elemente des Funktionsmodells dargestellt werden. In der Beschreibung des Metamodells wird deshalb der Bezug zwischen Funktionen und Vorgängen durch eine Zuordnung hergestellt.

Die Vorgänge werden in der ersten Spalte des VKD in Form einer Minispezifikation beschrieben. Die Vorgangsketten werden in den in der Mitte des VKD angeordneten Feldern, die eine Unterklasse zum Feld

Rasterung eingetragen. Eine Rasterung in zwei Bereiche entspricht den ursprünglichen VKD, eine mehrstufige Rasterung erlaubt eine klassifizierte Vorgangsdarstellung.

Die Spalten zur Darstellung der Datenbasis sind so angeordnet, daß die manuellen Datenbasen links und die DV-unterstützten Datenbasen rechts abgebildet werden. So können integrierte Datenspeicher über die gesamte VKD-Länge ohne zeichentechnische Probleme bei der Pfeilführung dargestellt werden.

Durch die Anordnung der Spalten bleibt auch bei den VOKAL-VKD eine Übersichtlichkeit der Darstellung erhalten. Der Betrachter erkennt den DV-Unterstützungsgrad der modellierten Vorgangskette aufgrund der Position der Vorgangs- und Datenbasissymbole. Dieses ist im Zusammenhang mit einer Kopplung des Systems VOKAL mit dem CIM-ANALYZER von Bedeutung.

Die Bilder 4a und 4b zeigen beispielhaft die VOKAL-VKD der Auftragsabwicklung im Ist- und im Sollzustand. Das Bild 4a zeigt den gleichen Modellierungsinhalt wie das Bild 2, um die Unterschiede zwischen der VOKAL-VKD-Struktur und der ursprünglichen VKD-Struktur aufzuzeigen.

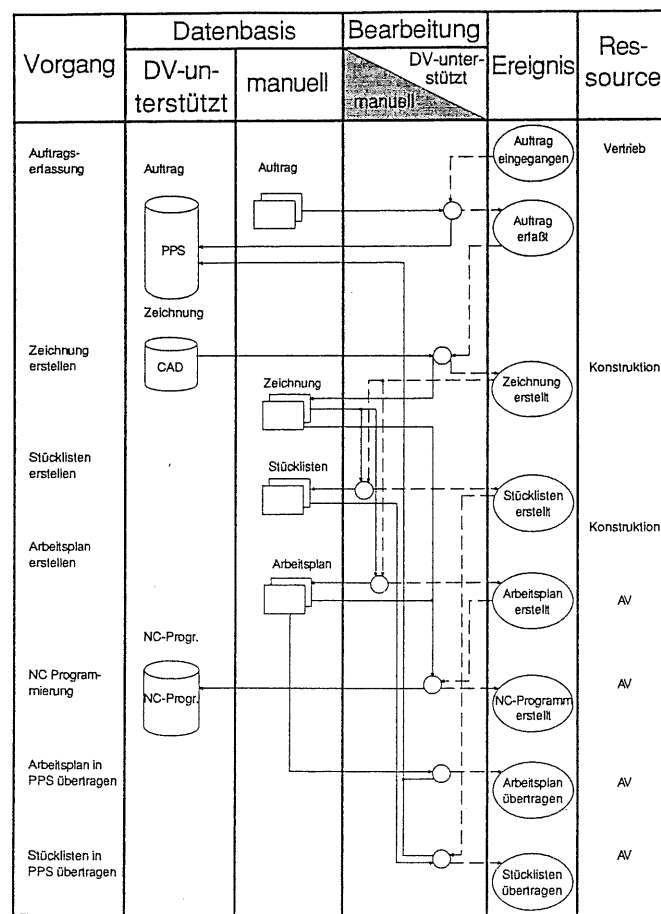


Bild 4a: VOKAL-VKD der Auftragsbearbeitung (Ist)



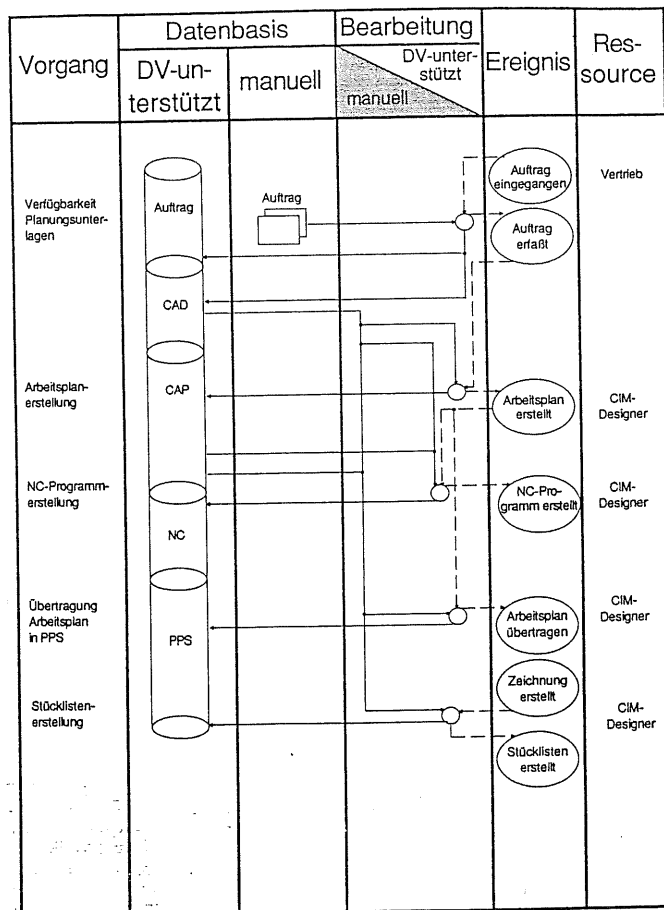


Bild 4b: VOKAL-VKD der Auftragsbearbeitung (Soll)

Die Struktur des VOKAL-VKD erlaubt eine übersichtliche und strukturierte Erfassung von Vorgangsketten bei gleichzeitiger leichter Verständlichkeit der Darstellung. Im Rahmen der auf die Erfassung nachfolgenden Vorgangskettenanalyse wird entsprechend des Analyseziels von dieser VKD-Struktur abgewichen. So ist beispielsweise eine Klassifizierung von einzelnen Vorgängen nach Durchlaufzeitdauern oder von Informationsflußbeziehung nach Frequenz als Analyseinhalt vorstellbar. Bei derartigen Analysen wird ausgehend von der in Bild 3 gezeigten VOKAL-VKD-Struktur abgewichen. Das VOKAL-VKD bleibt aber immer als Basis des Vorgangskettendiagramms bestehen.

Die Inhaltselemente, die in den beschriebenen Felder des VKD modelliert werden, sind unter dem Punkt 3.2 und 3.3 erläutert. Dort werden die Elemente der Methode zur Vorgangskettendarstellung und -analyse durch ein Metamodell dargestellt und anschließend attribuiert.

### 3.2 Metamodell des VKD

Die Methode zur Darstellung und Analyse von Vorgangsketten wird mit Hilfe eines Metamodells über VKD veranschaulicht. Das Metamodell, das die Beziehung der einzelnen Elemente des Vorgangskettenmodells

(VOKAL-VKD) zueinander in Beziehung setzt, ist in der Beschreibungssprache des Entity-Relationship-Modells (ERM) abgebildet. Es bildet die Grundlage zur Verknüpfung der Vorgangskettendiagramme mit anderen Beschreibungsmethoden.

Das Metamodell wird im folgenden zunächst nur für die inhaltlichen Elemente des VKD entwickelt und anschließend stufenweise um die Darstellung der Inhalte und um Verknüpfungsregeln zur Vorgangskettenmodellierung erweitert.

Die Beschreibung der Struktur des Metamodells folgt dieser Vorgehensweise. Es werden demzufolge zunächst die mit Hilfe der VOKAL-VKD darzustellenden Inhalte, anschließend deren Darstellung im VKD und zum Schluß die Konstruktionsregeln zur Verknüpfung der VKD-Elemente beschrieben.

Die Stabilität des Beschreibungsgegenstandes nimmt bei dieser Vorgehensweise sukzessive ab. Zeichnen sich die inhaltlichen Zusammenhänge der VKD-Elemente untereinander durch eine hohe Stabilität aus, ist die Darstellung von Inhalten vom Zweck der Darstellung abhängig und somit variabel. Im dargestellten Metamodell, das den Grundtyp von VKD abbildet, werden die VKD zum Zweck der Visualisierung der ablauforganisatorischen Zusammenhänge dargestellt. Sollen andere Darstellungsformen gewählt werden - wie beispielsweise eine Klassifizierung von Vorgängen nach deren Durchlaufzeit - kann die Struktur des VKD-Grundtyp durch eine modifizierte ersetzt werden. In diesem Fall baut das Modell zur Beschreibung der Darstellungsform auf dem Metamodell auf, das die darzustellenden Inhalte beschreibt. Die dritte Stufe der Methodenbeschreibung geht von der Darstellung der Inhalte in einem VOKAL-VKD (Grundtyp) aus und beschreibt die Verknüpfungsregeln der VKD-Elemente untereinander. Diese Beschreibungsform stellt die instabilste Form dar, weil sich sowohl die Darstellung selbst als auch die Konstruktionsregeln für die VKD-Elemente modifizieren lassen.

Aufgrund der stufenweisen Modellierungsbeschreibung kann bei jeder Änderung auf den stabilsten Beschreibungszustand zugegriffen werden. Es wird somit auch eine alternative Darstellungsform zu Vorgangskettendiagrammen ermöglicht. In diesem Fall bleibt die Beziehungsstruktur der Inhaltselemente bestehen, die Darstellungsform wird jedoch verändert. In Ergänzung zu den VKD können so Darstellungsalternativen zur Modellierung eingesetzt werden, wie sie beispielsweise bei Grochla aufgezeigt werden (Vgl. Grochla:1986). Der Inhalt und seine Strukturen, wie in Bild 5 dargestellt, bleiben bestehen, es erfolgt lediglich eine methodenspezifische Darstellungsveränderung.

In Bild 6 ist anschließend die Darstellung der beschriebenen Inhalte in VOKAL-VKD beschrieben. Durch Ergänzung des Bildes 6 können darüber hinaus Konstruktionsregeln zur Modellierung mit VKD abgebildet werden. Dieses wird in Bild 7 gezeigt werden.

Bei der ERM-Darstellung des VKD werden die Elemente der VKD durch Entities, die Beziehungen der einzelnen Elemente untereinander durch Relationen beschrieben. Entities werden im Entity-Relationship-Modell durch Rechtecke und Beziehungen durch Rauten dargestellt.

Der mit dem VKD darstellbare Inhalt kann durch die modellierten Elemente sowie durch die Ausprägung ihrer Attribute spezifiziert werden und bietet die Grundlage für die Entwicklung des Systems VOKAL. Die Kardinalitäten werden im Metamodell entsprechend der Schlageter/Stucky-Notation (Vgl. *Schlageter/Stucky*, 1983.) angegeben. Zur Unterscheidung von Entitytypen und deren Ausprägungen im Text werden die Entitytypen mit Großbuchstaben bei der Beschreibung des Metamodells gekennzeichnet. Beziehungstypen werden durch Anführungsstriche kenntlich gemacht.

Die Beschreibung des Metamodells bezieht sich im ersten Teil nur auf die abzubildenden Inhalte, wie sie in Bild 5 im Metamodell gezeigt werden.

Das Metamodell der Methode zur Darstellung und Analyse von Vorgangsketten geht von der VORGANGSKETTE als eigenständigem Entitytyp aus.

Eine VORGANGSKETTE stellt eine abgeschlossene Sequenz von Vorgängen dar. Der Entitytyp VORGANGSKETTE steht mit dem Entitytyp VORGANG in einer hierarchischen Beziehung. So wird gewährleistet, daß ein Vorgang immer vorgangskettenspezifisch modelliert wird. Eine Identifikation und redundanzfreie Speicherung von in verschiedenen Vorgangsketten enthaltenen Vorgängen kann somit nicht über die Identifikationsnummer des Vorgangs erfolgen, sondern kann nur durch Vergleich von Eingangs- und Ausgangsdatenbeziehungen sowie durch Gegenüberstellung der Minispezifikationen und der Vorgangsbezeichnungen durchgeführt werden. Die Gleichheit von Vorgängen wird allein durch die Äquivalenz von Datenbeziehungen und von Vorgangsbeschreibungen (MiniSpec) definiert.

Die Ausprägungen des VORGANGS können sowohl Bearbeitungen im Sinne einer Transformation von Eingangs- zu Ausgangsgrößen, Übertragungen als auch Entscheidungen sein. Weil die Attribute der Entitytypen ENTSCHEIDUNG, ÜBERTRAGUNG und BEARBEITUNG gleich sind, werden die ENTSCHEIDUNG, die ÜBERTRAGUNG und die BEARBEITUNG im Metamodell nur durch das Attribut Vorgangsart unterschieden.

Die VORGÄNGE werden durch eine sequenzorientierte "Vorgangs-Struktur" zueinander in Beziehung gesetzt. Diese gibt die Vorgänger-Nachfolger-Beziehungen zwischen den einzelnen vorgangskettenspezifischen Vorgängen an. Weil ein Vorgang sowohl mehrere Vorgänger als auch mehrere Nachfolger haben kann, wird die Vorgangsstruktur als eine n:m-Beziehung im Metamodell repräsentiert. Eine Rekursion innerhalb der Vorgangssequenz ist mit Hilfe dieser Struktur darstellbar.

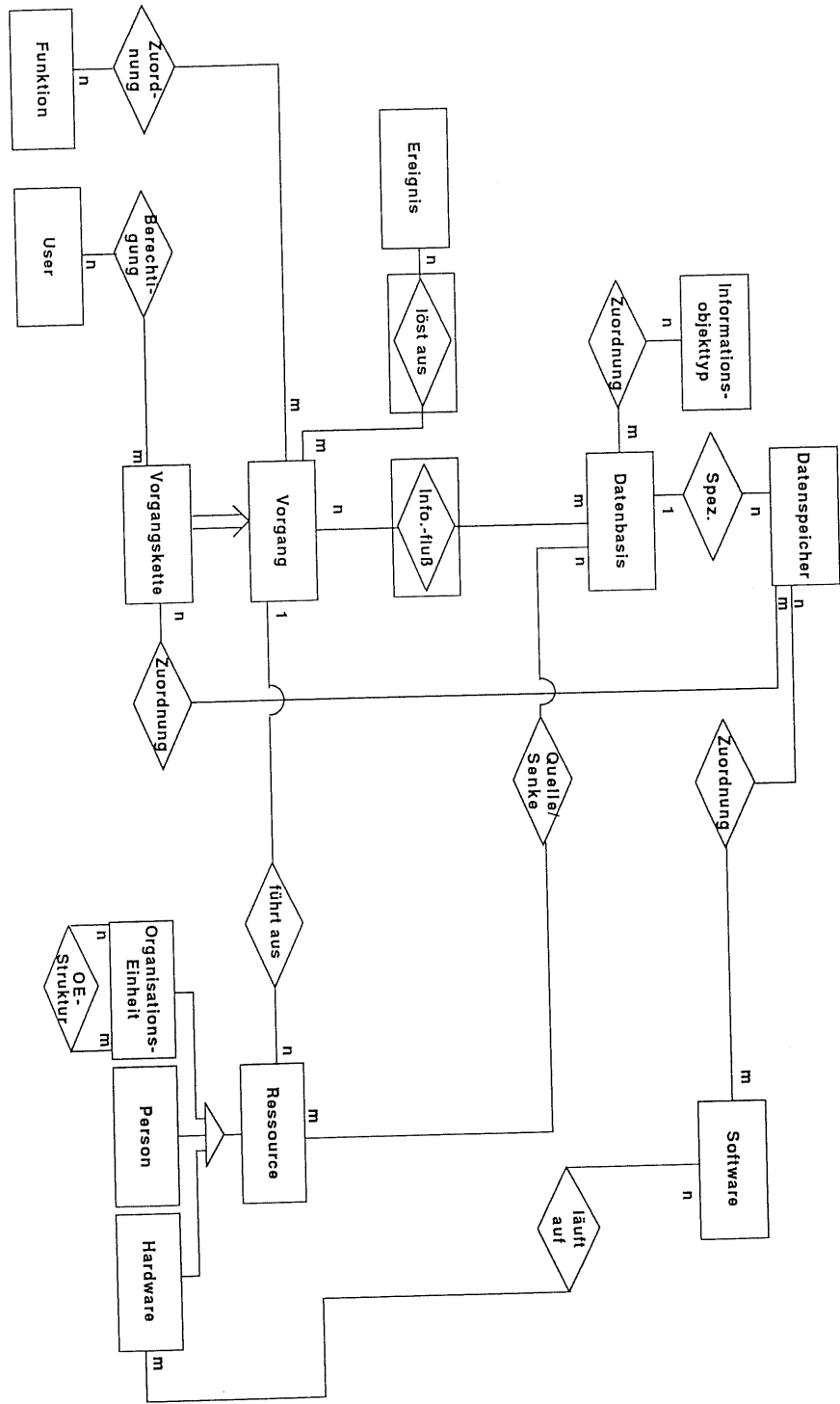


Bild 5: Metamodell der im VKD darzustellenden Inhalte

Weil die VKD eine Vorstufe zur fachkonzeptionellen Funktionsmodellierung darstellen, wird im Metamodell des VKD auch der Bezug zur Funktionsmodellierung hergestellt. Deshalb werden einem VORGANG FUNKTIONEN des Funktionsmodells zugewiesen. Weil die Abgrenzung zwischen Vorgängen und Funktionen nicht überschneidungsfrei ist, besteht zwischen VORGANG und FUNKTION eine n:m-Beziehung. Einem Vorgang können mehrere Funktionen zugewiesen werden, einer Funktion mehrere Vorgänge. Eine Funktion kann in unterschiedlichen Funktionsmodellen unterschiedlichen semantischen Inhalt haben. Sie wird deshalb durch die Schlüssel Funktionsbezeichnung und Funktionsmodellbezeichnung identifiziert.

Jedem Vorgang wird die ausführende Ressource zugeordnet. RESSOURCEN können sein: ORGANISATORISCHE EINHEITEN, PERSONEN oder HARDWARE-Systeme. Dieses wird durch die Spezialisierung des Entitytyp RESSOURCE in die Spezialisierungen ORGANISATORISCHE EINHEIT, PERSON und HARDWARE ausgedrückt.

Der Beziehungstyp "führt aus", der die Informationsobjekte VORGANG und RESSOURCE miteinander verbindet, gibt die Vorgangsausführung wieder. Dabei wird einem VORGANG immer genau eine RESSOURCE zugewiesen. Wird ein Vorgang durch mehreren Ressourcen ausgeführt, wird der Vorgang solange aufgeteilt, bis oben genannte Bedingung erfüllt ist. Diese Bedingung wird durch die dem Beziehungstypen "führt aus" zugeordnete Kardinalität 1:n dargestellt. Somit kann einem Vorgang immer nur eine Ressource zugewiesen werden, eine Ressource kann aber mehrere Vorgänge ausführen.

Eine ORGANISATIONSEINHEIT als Spezialisierung einer RESSOURCE stellt entsprechend des modellierten Detaillierungsgrades eine Stelle, eine Abteilung oder auch ein ganzes Werk dar. Die Organisationseinheiten stehen untereinander in einer Strukturbeziehung "OE-Struktur". Diese OE-Struktur ist als eine n:m-Beziehung dargestellt, weil die Einordnung der organisatorischen Einheiten in der Praxis nicht allein ausschließlich hierarchischen Kriterien folgt.

Ein weiteres zentrales Element von VKD stellen die Datenbanken dar. Sie werden im Metamodell durch den Entitytyp DATENBASIS abgebildet und stellen das Format bzw. das Dokument dar, mit dem die für Vorgänge relevanten und aus Vorgängen resultierenden Informationen gespeichert werden. Beispiele für Datenbanken sind ein Kundenauftrag, ein Wareneingangsschein, etc.. Die Datenbasis beschreibt nicht die physikalische Speicherung der Informationen (Vgl. Datenspeicher!). Eine Datenbasis als Element eines VKD stellt die Grundlage zur Ermittlung von Informationsobjekten im Rahmen der fachkonzeptionellen Datenmodellierung dar. Um die Ergebnisse der Vorgangskettenmodellierung in die Datenmodellierung einfließen lassen zu können, wird eine Beziehung zwischen der DATENBASIS und den INFORMATIONSOBJEKTEN eines Datenmodells modelliert. Ein Informationsobjekt kann in mehreren Datenmodellen enthalten sein und hat entsprechend unterschiedliche semantische Bedeutung. Aus diesem Grund wird das Informationsobjekt durch die Informationsobjektbezeichnung und die Bezeichnung des Datenmodells identifiziert. Die "Zuordnung" zwischen dem INFORMATIONSOBJEKT und der DATENBASIS ist vom Typ n:m. Es können einer

Datenbasis mehrere Informationsobjekte und einem Informationsobjekt mehrere Datenbasen zugewiesen werden.

Die DATENBASIS wird durch einen DATENSPEICHER spezifiziert. Der Datenspeicher gibt wieder, in welcher physikalischen Form eine Datenbasis abgespeichert wird. Ausprägungen eines DATENSPEICHERS können sein: manuelle Unterlagen, Datei auf Diskette, Datei auf Plattenspeicher, etc.. Die "Spezifikation" zwischen DATENSPEICHER und DATENBASIS ist vom Typ 1:n. Somit kann nur ein Datenspeicher einer Datenbasis zugeordnet werden. Prinzipiell könnten mehrere Datenspeicher einer Datenbasis (n:m-Beziehung) zugewiesen werden, dieses würde allerdings zu Darstellungsproblemen führen. Soll im VKD ausgedrückt werden, daß mehrere Datenspeicher verwendet werden, müssen demzufolge auch entsprechend viele Datenbasen modelliert werden.

Der DATENSPEICHER steht mit dem Entitytyp SOFTWARE in einer Zuordnungsbeziehung. Diese "Zuordnung" drückt aus, daß ein Datenspeicher durch eine oder mehrere Softwaresystem verwaltet werden kann. Weil mehrere Datenspeicher durch ein Softwaresystem verwaltet werden können und ein Softwaresystem verschiedene Datenspeicher verwalten kann, ist die Zuordnung vom Typ n:m. Die SOFTWARE ist mit der HARDWARE, die eine Spezialisierung der RESSOURCE darstellt durch die n:m-Beziehung "läuft auf" verbunden.

Eine Datenbasis ist mit einem Vorgang durch einen gerichteten "Informationsfluß" verbunden. Weil ein Informationsfluß eine Datenbasis mit mehreren Vorgängen verbinden kann und einem Vorgang durch den Informationsfluß mehrere Datenbasen zugewiesen werden können, ist die Beziehung zwischen DATENBASIS und VORGANG vom Typ n:m. Dem Informationsfluß wird als beschreibendes Attribut die Richtung zugewiesen. Ein zweiseitiger Informationsfluß zwischen DATENBASIS und VORGANG muß demzufolge durch zwei Informationsflüsse unterschiedlicher Richtung dargestellt werden.

Eine DATENBASIS kann einer RESSOURCE zugewiesen werden. Eine Beispiel für eine solche "Quelle/Senke"-Beziehung ist eine Verteilung von Dokumenten. Dabei ist das Dokument die Datenbasis, die noch um die physikalische Speicherung spezifiziert ist, und die Ressource ist der Empfänger des Dokuments. Weil die Verteilung von Dokumenten im VKD nicht als ein eigenständiger Vorgang betrachtet wird, wird sie als Quelle/Senke-Beziehung im Metamodell abgebildet. Die "Quelle/Senke" ist vom Typ n:m. Einer Ressource können mehrere Datenbasen zugewiesen werden und eine Datenbasis kann an mehrere Ressourcen verteilt werden. Die Quelle/Senke-Beziehung ist vom Informationsfluß zwischen Datenbasis und Vorgang zu differenzieren. Im VKD wird dieses durch unterschiedliche Darstellungen des Informationsflusses und der Quelle/Senke-Beziehung visualisiert. Weiterer Unterschied im VKD ist, daß ein Informationsfluß vom Datenspeichersymbol ausgeht, eine Quelle/Senke-Darstellung von der Datenbasisbeschreibung ausgeht.

Um vorgangskettenorientierte Auswertungen von Datenspeichermedien durchführen zu können, wird die



Ein VKD besteht aus mehreren VKD-SPALTEN. Zwischen dem VKD und den VKD-SPALTEN besteht eine hierarchische Abhängigkeit. Die Identifikation der VKD-SPALTE erfolgt über die VKD-Identnummer (VKDNR) und die VKD-Spalten-Identnummer (VKDSNR). Den VKD-SPALTEN werden die Darstellungssymbole des VKD zugeordnet. Deshalb steht die VKD-SPALTE zu allen im VKD abgebildeten Symbolen in Beziehung.

Die für den Fall, daß eine VORGANGSKETTE in mehreren VKD abgebildet wird, notwendige Verbindung mehrerer VKD untereinander erfolgt über die Strukturbeziehung "Konnektor". Die Ausprägungen des Beziehungstypen stellt die Verbindung von VKD untereinander jeweils paarweise dar. Weil prinzipiell jedes VKD mit verschiedenen anderen VKD außer mit sich selbst verbunden werden kann, ist die Beziehung Konnektor eine n:m-Zuordnung. Ein Konnektor wird jeweils durch genau zwei VKD-Identnummern identifiziert. Es wird dabei angegeben, welches VKD von welchem anderen VKD abhängig ist. Eine Abhängigkeit ist immer dann gegeben, wenn in einem VKD auf ein anderes VKD verwiesen wird. In dem Fall ist der Informationsfluß, der zu der Schnittstelle im VKD führt, so gerichtet, daß er von dem übergeordneten VKD zu dem abhängigen weist. Eine Abhängigkeit zwischen VKD existiert immer nur auf einen Konnektor bezogen. Demzufolge kann ein VKD gegenüber einem VKD abhängig, gegenüber einem weiteren übergeordnet sein. Aus diesem Grund wird das Abhängigkeitsverhältnis bei dem Informationsobjekt Konnektor abgebildet und nicht als Attribut eines VKD modelliert. Die Abhängigkeit zwischen zwei VKD wird im Metamodell so dargestellt, daß die VKD-Identnummern als ÜVKDNR (= übergeordnete VKD-Identnummer) und als UVKDNR (= untergeordnete VKD-Identnummer) modelliert werden.

Der Beziehungstyp "Konnektor" wird zum Entitytyp uminterpretiert und über die Beziehung "Konnektordarstellung" mit dem KONNEKTORSYMBOL und der VKD-SPALTE verbunden. Damit ist festgelegt, welcher Konnektor durch welches Symbol in welcher VKD-Spalte abgebildet wird. Ein Konnektor wird prinzipiell mindestens zweifach in VKD (in verschiedenen VKD-Spalten) abgebildet. Dieses, weil er zum einen in dem übergeordneten und zum anderen in dem untergeordneten VKD dargestellt wird. Weil es darüber hinaus auf der Ausprägungsebene zwei Arten von Konnektorsymbolen gibt, zum einen das Quell- zum anderen das Senke-Konnektorsymbol, besteht die Beziehung Konnektordarstellung als eine n:m:p-Beziehung. Bei der "Konnektordarstellung" werden die Attribute Koordinaten abgelegt. Das Konnektorsymbol ist im VKD in die Darstellung der Vorgangskette integriert und beinhaltet alle Informationen, um auf andere VKD verweisen zu können. Ein Konnektorsymbol wird durch ein stilisiertes VKD mit einer Identnummer und einem Seitenverweis auf das korrespondierende Konnektorsymbol abgebildet.

Ein KONNEKTOR unterbricht einen INFORMATIONSFLUSS zwischen einem VORGANG zu einer DATENBASIS. Dieses wird durch die Beziehung "unterbricht" zum Ausdruck gebracht. Die Beziehung bringt zum Ausdruck, daß die Unterbrechung sich auf den KONNEKTOR bezieht. Weil sinnvollerweise ein Informationsfluß durch genau einen Konnektor unterbrochen werden kann, ein Konnektor aber prinzipiell



mehrere Informationsflüsse unterbrechen kann, wird die Beziehung "unterbricht" als eine 1:n Beziehung modelliert. Kontrollflußdarstellungen oder Quelle/Senke-Darstellungen können nicht unterbrochen werden. Bei einem Seitenumbruch führt dieses zu einer redundanten Darstellung von Ereignissen, wenn das aus einem Vorgang resultierende Ereignis für einen nachfolgenden, in einem nachfolgenden VKD modellierten Vorgang auslösend wirkt.

Ein Vorgang wird im VKD durch eine Minispezifikation (MiniSpec) und ein VORGANGSSYMBOL dargestellt. Dazu besteht zwischen den Informationsobjekten VORGANG, und VKD-SPALTE eine Beziehung "MiniSpec" sowie zwischen dem VORGANG, dem VORGANGSSYMBOL und der VKD-SPALTE die Beziehung "Vorgangsdarstellung". Die Darstellung des Vorgangs (Vorgangssymbol) und der MiniSpec im VKD erfolgen auf gleicher Höhe, d. h. die Y-Koordinaten beider Darstellungen sind identisch.

Der Beziehungstyp "MiniSpec" gibt die definierende Beschreibung des jeweiligen Vorgangs in Kurzform wieder. Die Vorgangsbeschreibung wird im VKD in der Spalte mit der Bezeichnung Vorgänge aufgeführt. Sie wird um eine Langbeschreibung für den Vorgang ergänzt, die nicht standardmäßig im VKD abgebildet wird. Der Aufruf der Langbeschreibung zu einem Vorgang ist im VKD durch eine Buttonfunktion geplant. Weil in einer VKD-Spalte mehrere MiniSpecs aufgeführt werden können, und die MiniSpecs vorgangskettenspezifisch in verschiedenen VKD-Spalten (verschiedener VKD) abgebildet werden, besteht zwischen dem VORGANG und der VKD-SPALTE die n:m-Beziehung "MiniSpec".

Die Darstellung eines Vorgangs in den VKD-Spalten der Bezeichnung Bearbeitung kann prinzipiell in mehreren möglichen Spalten, je nach DV-Unterstützungsgrad, und durch verschiedene Vorgangssymbole erfolgen. Ein Vorgang selbst kann in einer Vorgangskette redundant modelliert werden. Somit ergibt sich die Beziehung "Vorgangsdarstellung" als eine n:m:p-Beziehung.

Eine Ressource wird als eine Beziehung zwischen den Entitytypen RESSOURCE und der VKD-SPALTE als n:m-Beziehung "Ressourcenbeschreibung" modelliert. Die Kardinalität des Beziehungstypen "Ressourcenbeschreibung" resultiert aus der Tatsache, daß eine Ressource in verschiedenen VKD-Spalten (verschiedener VKD) dargestellt werden kann. Gleichzeitig können in einer VKD-Spalte mehrere Ressourcen abgebildet werden.

Der INFORMATIONSFLUSS wird im VKD durch eine "Informationsflußdarstellung" dargestellt. Diese wird durch den Beziehungstyp "Infofluß" zwischen dem INFORMATIONSFLUSS und der VKD-SPALTE ausgedrückt. Die Darstellung eines Informationsflusses kann in mehreren Spalten modelliert werden. Ebenso kann eine VKD-Spalte mehrere Informationsflußdarstellungen beinhalten. Demzufolge besteht zwischen dem INFOFLUSS und der VKD-SPALTE eine n:m-Beziehung "Infoflußdarstellung".

Die DATENBASIS wird im VKD durch einen Beziehungstyp "Datenbasisbeschreibung" dargestellt. Die

Datenbasisbeschreibung verbindet die Entitytypen DATENBASISTEXT, VKD-SPALTE und DATENBASIS. Weil eine Datenbasis durch verschiedene Datenbasistexte und in verschiedenen VKD-Spalten modelliert werden kann, ist die Beziehung vom Typ 1:n:m.

Der DATENSPEICHER wird im VKD durch die "Datenspeicherdarstellung" mit der VKD-SPALTE und dem Datenspeichersymbol in Bezug gesetzt. Weil ein Datenspeicher in mehreren VKD-Spalten eingetragen werden kann aber nur durch ein Datenspeichersymbol repräsentiert wird, besteht eine 1:n:m Beziehung zwischen dem DATENSPEICHER, dem DATENSPEICHERSYMBOL und der VKD-SPALTE.

Ein EREIGNIS wird im VKD durch die Ereignisdarstellung repräsentiert. Dabei wird der Ereignisname in ein Ereignissymbol eingetragen. Weil nur ein Ereignissymbol existiert, wird im Metamodell kein eigenständiger Entitytyp EREIGNISSYMBOL eingeführt. Es besteht nur eine Beziehung zwischen EREIGNIS und VKD-SPALTE. Diese ist vom Typ n:m, weil ein Ereignis in mehreren VKD-Spalten (verschiedener VKD) abgebildet werden kann und eine VKD-Spalte mehrere Ereignisse enthalten kann.

Die Beziehung zwischen VORGANG und EREIGNIS "löst aus/ist Ergebnis von" wird zum Entitytyp uminterpretiert und mit der VKD-SPALTE in eine n:m Beziehung "Kontrollflußdarstellung" gesetzt. Der Kontrollfluß wird mehreren VKD-Spalten (Anfangs- und Endpunkt) zugewiesen, gleichzeitig können in einer VKD-Spalte mehrere Kontrollflüsse modelliert sein.

Die Darstellung der Quelle/Senke-Beziehung erfolgt durch die Quelle/Senke-Darstellung im VKD. Dieses ist eine n:m-Beziehung zwischen QUELLE/SENKE und der VKD-SPALTE, Weil eine Quelle/Senke-Darstellung in verschiedenen Spalten erfolgt (Anfang, Ende) und in einer VKD-Spalte verschiedene Quelle/Senke-Darstellungen enthalten sein können.

Somit sind alle Elemente des VKD zur Modellierung der in Bild 5 gezeigten Objekte beschrieben worden. Das in Bild 6 dargestellte Metamodell beinhaltet alle darzustellenden Inhalte sowie deren Abbildungselemente in einem Vorgangskettendiagramm.

Das Metamodell kann darüber hinaus um Konstruktionsregeln erweitert werden. Dieses Vorgehen bedeutet eine Einschränkung der VKD-Erstellungsmöglichkeiten. Dieses ist erforderlich, um im VKD nur sachlogisch richtige Einträge zu erzwingen. Prinzipiell können diese Einschränkungen ebenso durch Modellierungsregeln für das VKD beschrieben werden, die ihrerseits als explizite Regeln formuliert sind oder in Diagrammform (Matrizen) abgebildet werden. An dieser Stelle soll so verfahren werden, daß eine maximale Anzahl an Konstruktionsregeln aus dem Metamodell extrahiert werden kann. Somit hat man die Möglichkeit, die Methode zur Darstellung von Vorgangsketten mit geringem Beschreibungsaufwand semiformal und eindeutig darzustellen. Sollte es zur Veranschaulichung der Konstruktionsregeln erforderlich werden, zusätzliche Regeln zu formulieren bzw. Regeln in weiteren Darstellungen abzubilden, wird dieses ergänzend zu der Beschreibung

im Metamodell erfolgen. Dieses ist immer dann erforderlich, wenn sich die Regeln auf die Ausprägungsebene beziehen.

Für die skizzierte Erweiterung des Metamodells ausgehend von Bild 6 sei auf das Bild 7 verwiesen. Dieses stellt eine entsprechende Erweiterung des Metamodells um restriktive Konstruktionsregeln dar. Es wird im weiteren beschrieben.

Die in Bild 7 eingebrachten Erweiterungen beziehen sich im besonderen auf die Verknüpfung von Darstellungselementen des VKD. Es werden Regeln abgebildet, die das Zusammenwirken der einzelnen Symbole und Beschreibungen von der durch die Kombinatorik bestimmten Anzahl an Verknüpfungsmöglichkeiten auf ein sachlogisch sinnvolles Maß reduzieren.

Dieses geschieht dadurch, daß Beziehungstypen zwischen die Informationsobjekte des Metamodells, die Symbole oder Beschreibungen darstellen, eingefügt werden:

Zur Kennzeichnung, daß eine Informationsflußdarstellung immer nur mit einem Datenspeichersymbol verbunden werden darf, wird ein Beziehungstypen "Verbindung" als 1:n-Relation zwischen den uminterpretierten Beziehungstypen INFOFLUSS und DATENSPEICHERDARSTELLUNG modelliert. Dadurch wird definiert, daß die Informationsflußdarstellung niemals zwei Datenspeicherdarstellungen miteinander verbinden kann. Durch Einfügen eines weiteren Beziehungstypen "Verbindung" als 1:n-Relation zwischen den zum Entitytypen uminterpretierten Beziehungstypen VORGANGSDARSTELLUNG und INFOFLUSS wird analog reglementiert, daß eine Informationsflußdarstellung immer nur mit genau einer Vorgangsdarstellung verbunden werden kann. Eine Informationsflußdarstellung zwischen zwei Vorgangsdarstellungen ist somit unzulässig.

Weil eine Informationsflußdarstellung einen Informationsfluß - als Beziehung zwischen einer Datenbasis und einem Vorgang - im VKD repräsentiert, ergibt sich durch die beiden eingeführten "Verbindungen" eine alternierende, durch Informationsflußdarstellungen miteinander verbundene Reihenfolge zwischen Vorgangsdarstellungen und Datenspeicherdarstellungen. Dabei ist zulässig, daß von einer Datenspeicherdarstellung bzw. von einer Vorgangsdarstellung mehrere Informationsflußdarstellungen (Pfeile) ausgehen.



Eine Datenspeicherdarstellung steht mit der Datenbasisbeschreibung in einer 1:n Beziehung "Verbindung". Dieses, weil zu jeder Datenbasisbeschreibung nur genau eine Datenspeicherdarstellung zugeordnet werden darf. Einer Datenspeicherdarstellung können aber auf Typebene durchaus mehrere Datenbasisbeschreibungen zugewiesen werden.

Wird ein Informationsfluß durch einen Konnektor 'unterbrochen', so wird die Informationsflußdarstellung mit der Konnektordarstellung verbunden. Dieses erfolgt über den Beziehungstyp "Verbindung". Dabei können jeder Konnektordarstellung aus Gründen der Eindeutigkeit nur eine Informationsflußdarstellung (Pfeil) zugeordnet werden. Prinzipiell kann eine Informationsflußdarstellung mit mehreren Konnektordarstellungen verbunden sein. Der Beziehungstyp ist vom Typ 1:n. Eine Verbindung der Konnektordarstellung mit anderen Flußarten (Kontrollfluß, Quelle/Senke-Darstellung) ist nicht vorgesehen.

Zur Gewährleistung der Übersichtlichkeit der VKD wird der Beziehungstyp "Zuordnung" zwischen der QUELLE-/SENKE-DARSTELLUNG und der RESSOURCENBESCHREIBUNG als eine 1:n Verbindung eingeführt. So kann eine Quelle-/Senke-Darstellung zwar mit mehreren Ressourcenbeschreibungen verbunden werden, eine Ressourcenbeschreibung aber jeweils nur mit einer Quellen-/Senken-Darstellung.

Die QUELLE/SENKE-DARSTELLUNG ist darüber hinaus mit der DATENBASISBESCHREIBUNG verbunden. Der Beziehungstyp "Verbindung" ist vom Typ 1:n, weil eine Datenbasisbeschreibung mit mehreren Quelle/Senke-Darstellungen verbunden sein kann, eine Quelle/Senke-Darstellung jedoch jeweils nur mit einer Datenbasisbeschreibung verknüpft ist. Es ist zu beachten, daß im Gegensatz zur Informationsflußdarstellung die Quelle/Senke-Darstellung von der Datenbasisbeschreibung und nicht vom Datenspeichersymbol ausgeht.

Aufgrund der beschriebenen Regeln ist die Modellierung von VKD nunmehr komplett beschrieben. Es sind alle darzustellenden Inhalte und ihre Beziehungen sowie deren Darstellungen im VKD als auch die Regeln zur Modellierung von Vorgangsketten im Metamodell abgebildet. Im weiteren werden Definitionen der einzelnen Entity- und Beziehungstypen aufgeführt, die jeweils um die Attribute ergänzt werden.

### 3.3 Elementdefinition und Attributierung der VKD-Elemente

Aus dem Metamodell und dessen Beschreibung gehen die Regeln zur Verwendung der Elemente des VKD hervor. An den Stellen, an denen eine ERM-Darstellung der Methode auf der Typebene zur Methodenbeschreibung nicht ausreichend ist, werden die Beziehungen auf der Ausprägungsebene spezifiziert. (Vgl. Kap. 3.5.)

Die fachliche Spezifikation des Inhalts der einzelnen Informationsobjekte, die für die Methodenbeschreibung notwendig sind, ergeben sich aus den Entitytypen zuzuordnenden Attributausprägungen. Die Attribute werden deshalb bei der Beschreibung der Elemente des Metamodells ergänzend zu den Definitionen der Objekte aufgeführt.

Die Beschreibung der Elemente des Metamodells erfolgt demzufolge immer in der Reihenfolge: Objektbezeichnung, Definition, Schlüsselattribute und beschreibende Attribute.

Im folgenden werden die als Entity- oder Beziehungstyp im Metamodell dargestellten Elemente von VKD in alphabetischer Reihenfolge definiert und attribuiert.

## ABBILDUNG

Die Abbildung ist eine Zuordnung zwischen einem VKD und einer Vorgangskette. Sie definiert, welche Vorgangskette mit welchem VKD dargestellt wird. Dabei wird in der Regel eine Vorgangskette durch mehrere VKD abgebildet. Eine VKD enthält aber immer nur eine Vorgangskette.

### Identifizierende Attribute:

- VKDNR (= VKD-Identnummer)
- VKNR (= Vorgangsketten-Identnummer)

### Beschreibende Attribute:

- /

### **BERECHTIGUNG**

Eine Berechtigung ist ein Beziehungstyp zwischen einer Vorgangskette und einem USER. Sie definiert die Zugriffsrechte auf eine Vorgangskette und die Berechtigung Zugriffsrechte zu vergeben.

Beim Anlegen einer Vorgangskette erhält derjeniger Benutzer, der das Anlegen durchführt, im System VOKAL automatisch Projektleiterfunktion. Diese Berechtigung autorisiert ihn Benutzerrechte auf die Vorgangskette zu vergeben und zu entziehen. Die Berechtigung wird über die USER-ID und die VKNR (Vorgangskettennummer) identifiziert und ist demzufolge vorgangskettenspezifisch. Beim Abspeichern einer Vorgangskette, beispielsweise auf eine Diskette wird das Informationsobjekt Berechtigung automatisch mitübertragen, so daß der Benutzerzugriff auf eine Vorgangskette auch auf unterschiedlichen VOKAL-Systeme funktioniert.

#### **Identifizierende Attribute:**

- USER-ID
- VKNR

#### **Beschreibende Attribute:**

- angelegt am ...
- Projektleiter/Benutzer-Kennung
- Lese-/Schreib-Berechtigung
- Berechtigung von... bis...



## DATENBASIS

Eine Datenbasis beschreibt das Objekt oder das Format, dem Informationen gespeichert werden. Eine Datenbasis kann nur durch eine Funktion verändert werden.

Beispiele für Datenbasen sind ein Auftrag, eine Bestellung, etc.

### Identifizierende Attribute:

- DBNR (= Datenbasis-Identnummer)

### Beschreibende Attribute:

- Bezeichnung
- Format
- Zustand

## DATENBASISTEXT

Ein Datenbasistext beschreibt eine Datenbasis im VKD. Der Datenbasistext dient als einzige Datenbasisdarstellung und sollte durch einen Begriff die Datenbasis vollständig spezifizieren.

### Identifizierende Attribute:

- DBTNR (= Datenbasistext-Identnummer)

### Beschreibende Attribute:

- Sprache
- max. Zeichenlänge

## DATENBASISBESCHREIBUNG

Die Datenbasisbeschreibung ist eine Zuordnung zwischen der Datenbasis, einem Datenbasistext und einer VKD-Spalte.

Weil der Datenbasistext nicht zur vollständigen Beschreibung einer Datenbasis ausreicht, wird der Datenbasisbeschreibung ein erläuternder Beschreibungstext als Attribut hinzugefügt.

### Identifizierende Attribute:

- DBNR (= Datenbasis-Identnummer)
- DBTNR (= Datenbasistext-Identnummer)
- VKDNR (= VKD-Identnummer)
- VKDSNR (= VKD-Spalten-Identnummer)

### Beschreibende Attribute:

- X-Koordinate
- Y-Koordinate
- Feldlänge
- Datenbasisbeschreibungstext (erläuternd)

Beziehung "Zuordnung" zwischen dem DATENSPEICHER und der VORGANGSKETTE eingeführt. Es ist eine reine Zuordnung. Die Beziehung ist vom Typ  $n:m$ , weil ein Datenspeicher in verschiedenen Vorgangsketten vorkommen kann und gleichzeitig in einer Vorgangskette verschiedene Datenspeicher modelliert werden können.

VORGÄNGE werden durch EREIGNISSE ausgelöst. Ein Ereignis, das im VKD dargestellt wird, beschreibt einen temporären Umfeldzustand, der die Ausführung eines Vorgangs triggert oder als Ergebnis aus einem Vorgang resultiert. Nach Scheer (Vgl. Scheer, 1991) können mehrere Ereignisse einen Vorgang auslösen und ein Vorgang kann durch mehrere Ereignisse ausgelöst werden. Ähnliches gilt für die Beziehung "ist Ergebnis". Dem Beziehungstyp "löst aus/ ist Ergebnis" wird deshalb die Kardinalität  $n:m$  zugewiesen.

Das in Bild 5 dargestellte Metamodell der mit der Methode abzubildenden Inhalte ist an dieser Stelle vollständig beschrieben. Das entwickelte Metamodell kann als Ausgangsbasis für die Metamodellierung von unterschiedlichen Beschreibungsmethoden zur Vorgangskettendarstellung verwendet werden.

Im folgenden wird das beschriebene Metamodell zur Darstellung von Vorgangsketten mit Hilfe von VOKAL-VKD erweitert. Dazu werden die im Metamodell in Bild 5 beschriebenen Informationsobjekte zu den Elementen des VOKAL-VKD in Bezug gesetzt. Das resultierende Metamodell ist in Bild 6 abgebildet. Es ist eine Erweiterung des Bildes 5.: Das nunmehr vollständige Metamodell für die Modellierung und Analyse von Vorgangsketten wird im weiteren erläutert. Dabei wird auf die in Bild 6 dargestellten und aus Bild 5 übernommenen Inhalte nicht explizit eingegangen. (Zur leichteren Verständlichkeit der Beschreibung werden im folgenden einige Beziehungstypen mit ihrer Kurzbezeichnung beschrieben. Dieses sind namentlich die Beziehungstypen Zuordnung und Verbindung. Die Nomenklatur des Beziehungstypen schließt dabei die Entitytypen, zwischen denen die Verbindung besteht, mit ein, wie aus der nachfolgenden Beschreibung der Entity- und Beziehungstypen hervorgeht.)

Das Metamodell zeigt, daß eine VORGANGSKETTE durch ein VKD abgebildet wird. Dieses resultiert aus der Beziehung "Abbildung", die zwischen den Entitytypen VORGANGSKETTE und VKD besteht. Eine VKD beschreibt ein einzelnes Diagramm, das sich auf einem Blatt DIN-A4-Formats abbilden läßt und durch eine VKD-Identnummer identifiziert wird. Das Schlüsselattribut eines VKD enthält somit die Seitenzahl des VKD. Eine VORGANGSKETTE wird entsprechend ihrer durch die Anzahl von Vorgängen determinierten Länge in der Regel durch mehrere VKD abgebildet. Demzufolge läßt sich die Kardinalität des Beziehungstypen "Abbildung" als eine  $1:n$ -Beziehung beschreiben. Somit wird einem VKD immer genau eine VORGANGSKETTE zugeordnet, so daß die Abbildung von mehreren VORGANGSKETTEN in einem Diagramm ausgeschlossen wird. Gleichzeitig werden einer VORGANGSKETTE mehrere VKD zugeordnet, wenn die Komplexität oder der Umfang der abzubildenden Vorgangskette dieses erforderlich machen.

## DATENSPEICHER

Ein Datenspeicher beschreibt eine Datenbasis bezüglich der Form der Informationsspeicherung. Als Datenspeicher gilt jedes Medium, das Informationen für einen diskreten Zeitraum speichern kann. Weil ein Datenspeicher die physische Speicherung eines Informationsinhaltes darstellt, ist er häufiger Änderungen unterworfen als die Datenbasis, die über die Zuordnung "Spezifizierung" mit der Datenbasis verbunden ist.

Mögliche Ausprägungen für einen Datenspeicher sind: Datei auf Diskette, Informationen auf manueller Unterlage, Plattenspeicher, etc.

Ein Datenspeicher kann ebenso eine Informationsübertragung beschreiben. In diesem Zusammenhang kann auch ein Telefon als Datenspeicher betrachtet werden, weil es für eine (zwar kurze) diskrete Zeitdauer Informationen speichert.

### Identifizierende Attribute:

- DSNR (= Datenspeicher-Identnummer)

### Beschreibende Attribute:

- Bezeichnung
- manuell/DV-unterstützt-Kennung
- Zustand
- Speicherkapazität

## DATENSPEICHERSYMBOL

Ein Datenspeichersymbol stellt einen Datenspeicher im VKD grafisch dar. Es wird als eigenständiger Entitytyp im Metamodell geführt und kann über die Beziehung "Datenspeicherdarstellung" mit dem Datenspeicher und der VKD-Spalte in Beziehung gesetzt werden. Der Entitytyp Datenspeichersymbol beinhaltet alle möglichen Datenspeichersymbole in nicht spezifizierter Form. Beispiele für Ausprägungen des Datenspeichersymbols sind: Symbol für eine Datei auf Diskette, Symbol für eine manuelle Unterlage, etc..

### Identifizierende Attribute:

- DSSNR (= Datenspeichersymbol-Identnummer)

### Beschreibende Attribute:

- Symbolart (z.B. "Datei auf Diskette", "Datei auf Bandspeicher")
- Symbolmaße
- X-Koordinate des Symbolmittelpunkts
- Y-Koordinate des Symbolmittelpunkts
- Flächeninhalt

## DATENSPEICHERDARSTELLUNG

Eine Datenspeicherdarstellung ist eine Zuordnung zwischen einem Datenspeicher, einem Datenspeichersymbol und einer VKD-Spalte.

### Identifizierende Attribute:

- DSNR (= Datenspeicher-Identnummer)
- DSSNR (= Datenspeichersymbol-Identnummer)
- VKDNR (= VKD-Identnummer)
- VKDSNR (= VKD-Spalten-Identnummer)

### Beschreibende Attribute:

- X-Koordinate
- Y-Koordinate
- Spezifizierung
- Spezifizierungstext

## **INFORMATIONSOBJEKTYP (IO)**

Ein Informationsobjekttyp ist ein Element eines Datenmodells auf der Typebene. Durch die Zuordnung zwischen dem Informationsobjekttyp und der Datenbasis wird eine Verbindung zwischen der fachkonzeptionellen Datenmodellierung und der Modellierung von Vorgangsketten hergestellt.

Weil in verschiedenen Datenmodellen die Informationsobjekttypen unterschiedliche Bedeutung haben können, wird das Datenmodell als identifizierendes Attribut des Informationsobjekttyp mit aufgeführt.

### **Identifizierende Attribute:**

- Informationsobjekttypbezeichnung
- Datenmodell

### **Beschreibende Attribute:**

- Schlüsselattribut
- /



## EREIGNIS/ERGEBNIS

Ein Ereignis/Ergebnis beschreibt einen für einen diskreten Zeitraum konstanten Zustand, der die Ausführung eines Vorgangs auslöst und/oder der das Ergebnis eines Vorgangs ist. In der Regel sind Ereignisse/Ergebnisse durch ein Informationsobjekt des Datenmodells und einen der Ausprägung dieses Informationsobjektes zugeordneten Zustands definiert. Weil jedoch nicht alle Ereignisse in dieser Form definierbar sind, wird das Ereignis als eigenständiger Entitytyp im Metamodell aufgeführt.

Ein Beispiel für ein Ereignis ist: "Kundenauftrag eingegangen".

Ereignisse stoßen Vorgänge an und werden von Vorgängen erzeugt.

### Identifizierende Attribute:

- ENR (= Ereignis-Identnummer)

### Beschreibende Attribute:

- Bezeichnung
- Beschreibung
- Ereignis/Ergebnis-Kennung

## EREIGNIS-/ERGEBNIS-DARSTELLUNG

Ein Ergebnis oder Ereignis werden im VKD gleichartig dargestellt. Ein Ereignis kann für einen Vorgang als ein auslösendes Ereignis für einen weiteren Vorgang als ein Ergebnis betrachtet werden.

Ereignisse und Ergebnisse werden im VOKAL-VKD in einer einzelnen Spalte "Ereignisse" abgebildet und durch Kontrollflußdarstellungen mit Vorgängen verbunden.

Die Funktionalität von VOKAL sieht ein Ausblenden der Ereignisspalte incl. der Ereignisse und Kontrollflüsse vor.

### Identifizierende Attribute

- ENR (= Ereignis-Identnummer)
- VKDNR (= VKD-Identnummer)
- VKDSNR (= VKD-Spaltennummer)

### Beschreibende Attribute:

- X-Koordinate
- Y-Koordinate
- Flächenattribute

### FUNKTION

Eine Funktion ist ein Element eines fachkonzeptionellen Funktionsmodells. Sie stellt den Bezug zwischen der Funktionsmodellierung und der Modellierung von VKD her. Weil eine Funktion in verschiedenen Funktionsmodellen unterschiedliche Bedeutung haben kann, wird die Funktionsmodellbezeichnung als identifizierendes Attribut der Funktion hinzugefügt.

#### **Identifizierende Attribute:**

- Funktionsbezeichnung
- Funktionsmodellbezeichnung

#### **Beschreibende Attribute:**

- /

**FÜHRT AUS:**

Die Zuordnung "führt aus" verbindet einen Vorgang mit einer Ressource. Weil der Vorgang selbst vorgangskettenspezifisch ist, wird auch der Beziehungstyp vorgangskettenspezifisch modelliert. Einem Vorgang wird immer eindeutig eine Ressource zugewiesen. Sollte ein Vorgang von mehreren Ressourcen ausgeführt werden, ist dieser solange aufzuteilen, bis Teilvorgänge entstehen, denen eindeutig eine ausführende Ressource zugeordnet werden kann. Sollte dieses nicht möglich sein, wird diejenige Ressource, die den Vorgang verantwortlich ausgeführt hat, mit dem Vorgang in Beziehung gesetzt. Dieses ist beispielsweise bei der Modellierung von Abstimmungsvorgängen die federführende Abteilung.

**Identifizierende Attribute:**

- VNR (= Vorgangs-Identnummer)
- VKNR (= VK-Identnummer)
- RNR (= Ressourcen-Identnummer)

**Beschreibende Attribute:**

- /

## **HARDWARE (HW)**

Hardware ist eine Spezialisierung des Entitytyp Ressource. Hardware kann ebenfalls Vorgänge ausführen.

### **Identifizierende Attribute:**

- RNR (= Ressourcen-Identnummer)

### **Beschreibende Attribute:**

- OE/Person/HW-Kennung
- Bezeichnung

## INFORMATIONSFLOSS (INFO-FLUSS)

Ein Informationsfluß ist der Informationstransport ausgehend von einem Datenspeicher und hinführend zu einem Vorgang bzw. umgekehrt. Der Informationsfluß hat eine definierte Richtung und eine Frequenzangabe. Weil ein Vorgang hierarchisch von der Vorgangskette abhängt, ist jeder Informationsfluß vorgangskettenspezifisch.

### Identifizierende Attribute:

- VNR (= Vorgangs-Identnummer)
- VKNR (= VK-Identnummer)
- DBNR (= Datenbasis-Identnummer)

### Beschreibende Attribute:

- Richtung ("+" = vom Vorgang zur Datenbasis; "-" von der Datenbasis zum Vorgang)
- Häufigkeit (Frequenz)
- Mengenangaben
- physikalische Realisierung des Datenflusses

## INFORMATIONSFLOSSDARSTELLUNG (Infoflußdarstellung)

Die Informationsflußdarstellung stellt im VKD einen Transport von Informationen grafisch dar. Durch die Definition des Informationsflusses als Beziehungstyp zwischen dem Vorgang und der Datenbasis ist bereits festgelegt, daß die Informationsflußdarstellung eine Datenbasisbeschreibung mit einer Vorgangsdarstellung verbindet. Die Informationsflußdarstellung wird durch einen gerichteten Pfeil abgebildet. Für den Pfeil werden die Anfangs- und Endkoordinaten, die mit den Randkoordinaten der verbundenen Symbole übereinstimmen, bestimmt. Es wird dafür ein spaltenübergreifendes kartesisches Koordinatensystem zugrunde gelegt. Der Weg des Pfeils durch das VKD ist inhaltlich irrelevant und muß demzufolge nur nach Übersichtlichkeitskriterien gewählt werden. Er wird vom System VOKAL selbständig erstellt.

### Identifizierende Attribute:

- VNR (= Vorgangs-Identnummer)
- VKNR (= Vorgangsketten-Identnummer)
- DSNR (= Datenspeicher-Identnummer)
- VKDNR (= VKD-Identnummer)
- AVKDSNR (= VKD-Spaltenidentnummer-Anfang)
- EVKDSNR (= VKD-Spalten-Identnummer-Ende)

### Beschreibende Attribute:

- Anfangs-X-Koordinate
- End-X-Koordinate
- Anfangs-Y-Koordinate
- End-Y-Koordinate

## KONNEKTOR

Ein Konnektor ist die Verbindung von zwei VKD. Durch die Konnektoren wird zwischen VKD eine Struktur aufgebaut, bei der zwischen einem übergeordneten VKD und einem abhängigen VKD unterschieden wird. Zwei VKD werden über Konnektoren miteinander verbunden, wenn eine Vorgangskette in einem VKD beginnt und in einem weiteren aus zeichentechnischen Gründen fortgesetzt werden muß. Fließt dabei der durch einen Pfeil im VKD repräsentierte Informationsfluß von einem VKD 'A' in ein VKD 'B', dann wird 'A' als das übergeordnete, 'B' als das abhängige VKD definiert.

Der Verweis auf ein weiteres VKD erfolgt sowohl in dem übergeordneten als auch in dem abhängigen VKD durch ein Konnektorsymbol.

### Identifizierende Attribute:

- ÜVKDNR (= übergeordnete VKD-Identnummer)
- UVKDNR (= untergeordnete VKD-Identnummer)

### Beschreibende Attribute:

- /



## KONNEKTORDARSTELLUNG

Eine Konnektordarstellung ist die Zuordnung zwischen Konnektor, Konnektorsymbol und VKD-Spalte. Es werden einem Konnektor in der Regel zwei Konnektorsymbole (eine Quelle und eine Senke) sowie für jedes Symbol die Koordinaten innerhalb der spezifizierten VKD-Spalte bestimmt.

Identifizierende Attribute:

- ÜVKDNR (= übergeordnete VKD-Identnummer)
- UVKDNR (= untergeordnete VKD-Identnummer)
- KONSNR (= Konnektorsymbol-Identnummer)
- VKDNR (= VKD-Identnummer)
- VKDSNR (= VKD-Spalten-Identnummer)

Beschreibende Attribute:

- Zusammengesetztes Attribut aus ÜVKDNR/UVKDNR
- X-Koordinate
- Y-Koordinate
- Seitenverweis

## KONNEKTORSYMBOL

Ein Konnektorsymbol stellt Konnektoren als eine Verbindungshilfe von VKD untereinander grafisch dar. Es wird als ein stilisiertes VKD dargestellt, das ein Feld für die Identifikationsnummer (= das zusammengesetzte Attribut aus ÜVKDNR/UVKDNR) sowie den Seitenverweis des in Beziehung stehenden VKD enthält.

Es existieren Quellen- und Senken-Konnektorsymbole.

### Identifizierende Attribute

- KONSNR (= Konnektorsymbol-Identnummer)

### Beschreibende Attribute:

- Flächenattribute
- Quelle/Senke-Kennung

## KONTROLLFLUSSDARSTELLUNG

Der Kontrollfluß stellt im VKD die Verbindung zwischen Ereignissen/ Ergebnissen und Vorgängen grafisch dar. Er wird als strichpunktierte Linie verdeutlicht, um ihn von den Informationsflußdarstellungen optisch unterscheidbar zu machen.

### Identifizierende Attribute:

- ENR (= Ereignis/Ergebnis-Identnummer)
- VKNR (= Vorgangsketten-Identnummer)
- VNR (= Vorgangs-Identnummer)
- VKDNR (= VKD-Identnummer)
- AVKDSNR (= VKD-Spalten-Identnummer-Anfang)
- EVKDSNR (= VKD-Spalten-Identnummer-Ende)

### Beschreibende Attribute:

- X-Koordinate Anfang
- Y-Koordinate Anfang
- X-Koordinate Ende
- Y-Koordinate Ende
- Linienattribut (Farbe, gestrichelt, ..)

## LÄUFT AUF

Die Beziehung **läuft auf** verbindet die modellierten Softwarekomponenten mit den Hardwarekomponenten.

### **Identifizierende Attribute:**

- SWNR (= Software-Identnummer)
- RNR (= Ressourcen-Identnummer)

### **Beschreibende Attribute:**

- /

## LÄUFT AUF

Die Beziehung läuft auf verbindet die modellierten Softwarekomponenten mit den Hardwarekomponenten.

### Identifizierende Attribute:

- SWNR (= Software-Identnummer)
- RNR (= Ressourcen-Identnummer)

### Beschreibende Attribute:

- /

## LÖST AUS/ IST ERGEBNIS VON

Die Beziehung löst aus/ ist Ergebnis von ordnet den Ereignissen Vorgänge zu, die durch diese ausgelöst werden. Darüber hinaus werden den Vorgängen die Ergebnisse zugewiesen, die aus ihnen resultieren.

### Identifizierende Attribute:

- VNR (= Vorgangs-Identnummer)
- ENR (= Ereignis-Identnummer)

### Beschreibende Attribute:

- Auslöser/Ergebnis-Kennung

## MINISPEZIFIKATION (MiniSpec)

Die Minispezifikation (MiniSpec) ist ein Vorgangsbeschreibungstext, der einem Vorgang und einer VKD-Spalte zugeordnet wird. Der Vorgang wird mit Hilfe der Minispec definiert. Durch die Minispezifikationen wird ein substantieller Beitrag zur Verständlichkeit von VKD geleistet.

Die MiniSpec zu einem Vorgang erscheint in der Spalte Vorgang des VKD.

### Identifizierende Attribute:

- VKNR (= Vorgangsketten-Identnummer)
- VNR (= Vorgangs-Identnummer)
- VKDNR (=VKD-Identnummer)
- VKDSNR (=VKD-Spalten-Identnummer)

### Beschreibende Attribute:

- X-Koordinate
- Y-Koordinate
- Flächenattribute

## ORGANISATIONSEINHEIT (OE-EINHEIT)

Eine Organisationseinheit ist eine Spezialisierung des Entitytypen Ressource. Es beschreibt eine Stelle und ihre Stellung innerhalb des Organisationsdiagramms. Eine Organisationseinheit kann je nach Detaillierungsgrad des VKD eine Stelle oder eine Abteilung sein.

### Identifizierende Attribute:

- RNR(= Ressourcen-Identnummer)

### Beschreibende Attribute:

- OE/Person/HW-Kennung
- Bezeichnung



## ORGANISATIONSEINHEIT-STRUKTUR (OE-STRUKTUR)

Die Struktur der Organisationseinheiten ist eine Zuordnungsstruktur. Es werden die Einordnung der Organisationseinheiten untereinander entsprechend des Organisationsdiagramms beschreiben. In der OE-Struktur ist festgelegt, welche Organisationseinheit welcher über- bzw. untergeordnet ist.

Die Organisationsstruktur bezieht sich nur auf Ressourcen vom Typ Organisatorische Einheit.

### Identifizierende Attribute:

- ORNR (= Ober-Ressourcen-Identnummer)
- URNR (= Unter-Ressourcen-Identnummer)

### Beschreibende Attribute:

- /

### PERSON

Eine Person ist eine Spezialisierung des Entitytyp Ressource.

#### Identifizierende Attribute:

- RNR (= Ressourcen-Identnummer)

#### Beschreibende Attribute:

- OE/Person/HW-Kennung
- Bezeichnung

---

## QUELLE/ SENKE

Im Falle einer Verteilung von Informationen als Vorgang einer Vorgangskette wird die Datenbasis mit der empfangenden Ressource in Bezug gesetzt. dieses wird im VKD durch einen Doppelpfeil dargestellt.

### Identifizierende Attribute:

- RNR (= Identnummer der organisatorischen Einheit)
- DBNR (= Datenbasis-Identnummer)

### Beschreibende Attribute:

- Anzahl der Empfänger
- Absender/Empfänger

## QUELLE-/ SENKE-DARSTELLUNG

Eine Quelle/Senke-Darstellung visualisiert die Verteilung von Datenbasen auf Ressourcen im VKD. Die Verteilung wird zur Unterscheidung vom Informationsfluß durch einen Doppelpfeil dargestellt. Die Quelle/Senke-Darstellung ist eine Beziehung zwischen einer Quelle/Senke und einer VKD-Spalte.

### Identifizierende Attribute:

- DBNR
- RNR
- VKDNR
- VKDSNR

### Beschreibende Attribute:

- Y-Koordinate (Höhe)

## RESSOURCE

Eine Ressource ist eine Generalisierung der Aufgabenträger Person, organisatorische Einheit und Hardware.

Eine Ressource ist in der Lage, Vorgänge selbständig auszuführen und Informationen zu empfangen.

### Identifizierende Attribute:

- RNR (= Ressourcen-Identnummer)

### Beschreibende Attribute:

- OE/Person/HW-Kennung
- Bezeichnung

## RESSOURCENBESCHREIBUNG

Das Ressourcenbeschreibung dient der Darstellung von Ressourcen im VKD. Eine Ressource wird durch einen Begriff im VKD dargestellt. Die Ressourcenbeschreibung wird im Metamodell als eine Zuordnung zwischen den Entitytypen Ressource und VKD-Spalte modelliert.

Es wird beschrieben, an welcher Stelle im VKD und durch welchen Begriff die Ressource dargestellt wird.

Weil eine Ressource an verschiedenen Orten im VKD abgebildet werden kann und die Bezeichnung vom Beziehungstyp Ressourcenbegriff abhängt, kann prinzipiell eine Ressource durch unterschiedliche Begriffe im VKD beschrieben werden. Es sollten jedoch für gleiche Ressourcen innerhalb einer Vorgangskette gleiche Bezeichnungen verwendet werden. Jede Ressourcenbeschreibung bezieht sich auf genau eine Ressource. Ein Abbilden eines Sammelbegriff für mehrere Ressourcen ist nicht vorgesehen. Dieses kann lediglich durch Modellierung einer ganzen Abteilung als Ressource erfolgen.

### Identifizierende Attribute:

- RNR (= Ressourcen-Identnummer)
- VKDNR (= VKD-Identnummer)
- VKDSNR (= VKD-Spalten-Identnummer)

### Beschreibende Attribute:

- Bezeichnung
- Bezeichnungslänge
- Beschreibung
- Beschreibungslänge
- X-Koordinate des Feldmittelpunkts
- Y-Koordinate des Feldmittelpunkts
- Flächenattribute
- Sprache

## SOFTWARE (SW)

Software repräsentiert die für die Vorgangskette relevanten Softwaresysteme. Im VKD wird Software nicht als ein System betrachtet, das Vorgänge ausführen kann. Die Hardware als Spezialisierung der Ressource kann Vorgänge ausführen. Eine Software läuft auf einer Hardwarekomponente und wird deshalb durch die Beziehung läuft auf der Hardware zugewiesen.

### **Identifizierende Attribute:**

- SWNR (= Software-Identnummer)

### **Beschreibende Attribute:**

- Bezeichnung
- Modulbezeichnung
- Hersteller

### SPEZIFIZIERUNG

Die Spezifizierung stellt eine Zuordnung zwischen einem Datenspeicher und einer Datenbasis dar. Es wird jeder Datenbasis wie beispielsweise einem Auftrag, Lieferschein, etc. die physikalische Datenspeicherung zugeordnet. So wird beispielsweise die Datenbasis Auftrag um die Datenspeicherung als manuelle Unterlage oder als EDI-Nachrichtentyp spezifiziert.

#### **Identifizierende Attribute:**

- DBNR (= Datenbasis-Identnummer)
- DSNR (= Datenspeicher-Identnummer)

#### **Beschreibende Attribute:**

- /



## UNTERBRICHT

Unterbricht ist eine Beziehung zwischen einem Konnektor und einem Informationsfluß. Der Informationsfluß zwischen einem Vorgang und einer Datenbasis wird durch einen Konnektor unterbrochen.

### Identifizierende Attribute:

- DBNR (= Datenbasis-Identnummer)
- VNR (= Vorgangs-Identnummer)
- VKNR (= VK-Identnummer)
- UVKDNR (= untergeordnetes VKD)
- ÜVKDNR (= übergeordnetes VKD)

### Beschreibende Attribute:

- /

### USER

Der User ist der Benutzer des Systems VOKAL: Er identifiziert sich gegenüber dem System VOKAL durch eine Benutzerkennung (USER-ID) und ein Password.

Beim erstmaligen Anmelden eines USER wird eine USER-ID und das angegebene Password beim USER angelegt und beim nachfolgendem Einloggen abgefragt. Wird sich beim Eingeben der USER-ID verschrieben, wird demzufolge ein neuer Benutzer incl. Password angelegt.

Diese Informationen werden im Benutzerverwaltungsmodul von VOKAL verwaltet.

Durch den Beziehungstyp Berechtigung wird jedem Benutzer bezüglich einer Vorgangskette eine Benutzerberechtigung zugewiesen. Diese Berechtigung, die vom Benutzer und der Vorgangskette abhängig ist, ist von der USER-Berechtigung, die dem USER direkt zugeordnet ist, zu differenzieren. In der USER-Berechtigung werden Systemfunktionen, wie beispielsweise das Löschen von Benutzern, das Verändern von Zugriffsberechtigungen etc. beschrieben.

Identifizierende Attribute:

- USER-ID

Beschreibende Attribute:

- Password
- Name
- USER-Berechtigung

## VERBINDUNG Datenspeicherdarstellung-Datenbasisbeschreibung

Eine Verbindung repräsentiert eine Konstruktionsregel des VOKAL-VKD.

Die Verbindung spezifiziert, welche Datenbasisbeschreibung durch welche Datenspeicherdarstellung ergänzt wird.

### Identifizierende Attribute:

- DSNR
- DSSNR
- VKDNR
- VKDSNR
- DBNR
- DBTNR
- VKDNR
- VKDSNR

### Beschreibende Attribute:

- /

## **VERBINDUNG Datenspeicherdarstellung Informationsflußdarstellung**

Eine Verbindung repräsentiert eine Konstruktionsregel des VOKAL-VKD.

Die Verbindung beschreibt, welche Datenspeicherdarstellung mit welcher Informationsflußdarstellung (Pfeil) verbunden wird.

### **Identifizierende Attribute:**

- DSNR
- DSSNR
- VKDNR
- VKDSNR
- DBNR
- VNR
- VKNR

### **Beschreibende Attribute:**

- X-Koordinate
- Y-Koordinate

## **VERBINDUNG Infoflußdarstellung-Vorgangsdarstellung**

Eine Verbindung repräsentiert eine Konstruktionsregel des VOKAL-VKD.

Sie drückt aus welche Vorgangsdarstellung mit welcher Informationsflußdarstellung verbunden wird. Weil die Verbindung an einem Punkt des VKD erfolgt, wird bei den beschreibenden Attributen der Verbindungspunkt beschrieben.

### **Identifizierende Attribute:**

- DBNR
- VNR
- VKNR
- VKDNR
- VKDSNR

### **Beschreibende Attribute**

- X-Koordinate
- Y-Koordinate

## **VERBINDUNG Konnektordarstellung Informationsflußdarstellung**

Eine Verbindung repräsentiert eine Konstruktionsregel des VOKAL-VKD.

Eine Konnektordarstellung stellt eine Informationsflußunterbrechung dar. Für den Informationsfluß stellt das Konnektorsymbol ein Schnittstellensymbol dar. Die Verbindung zwischen dem Konnektorsymbol und der Informationsflußdarstellung wird in dieser Verbindung beschrieben. Es existieren grundsätzlich zwei zueinander komplementäre Verbindung zwischen einer Informationsflußdarstellung (in verschiedenen VKD) und zwei einander zugeordneten Konnektordarstellungen.

### **Identifizierende Attribute:**

- UVKDNR
- ÜVKDNR
- KONSNR
- VKDSNR
- VKNR
- VNR
- DBNR

### **Beschreibende Attribute:**

- X-Koordinate
- Y-Koordinate

**VERBINDUNG Quelle/Senke\_Darstellung - Datenbasisbeschreibung**

Die Verbindung drückt aus welche Datenbasisbeschreibung mit welcher Quelle/Senke-Darstellung verbunden ist. Es wird ein Punkt im Koordinatensystem des VOKAL-VKD identifiziert, der beiden Darstellungen gemeinsam ist. Dieses ist der Verbindungspunkt.

**Identifizierende Attribute:**

- DBNR
- DBTNR
- VKDNR
- VKDSNR
- RNR

**Beschreibende Attribute:**

- X-Koordinate des Verbindungspunkts
- Y-Koordinate des Verbindungspunkts (Höhe)

## VORGANG

Ein Vorgang ist Element einer Vorgangskette. Er transformiert Eingangsgrößen zu veränderten Ausgangsgrößen, führt Entscheidungen aus oder transportiert Informationen. Ein Vorgang wird durch ein Ereignis angestoßen und erzeugt ein für einen diskreten Zeitraum konstantes Endergebnis. Kann das Resultat eines Vorgangs nicht identifiziert werden bzw. ist dieses nicht konstant, werden mehrere Teilvorgänge zu einem Gesamtvorgang zusammengefaßt, der dann im Vorgangskettendiagramm modelliert werden kann.

Im VKD wird abgebildet, welche Eingangsgrößen in Form einer Datenbasis in einen Vorgang eingehen und welche Ausgangsgrößen aus ihm resultieren. Darüber hinaus wird dargestellt welche Ereignisse einen Vorgang auslösen und welche Ergebnisereignisse aus dem Vorgang resultieren.

Ein Vorgang wird durch genau eine Ressource ausgeführt. Kommen mehrere Ressourcen zur Vorgangsausführung in Frage, muß der Vorgang zergliedert werden, bis oben genannte Bedingung erfüllt ist.

Ein Vorgang wird im Metamodell als hierarchisch abhängig von der Vorgangskette modelliert. Dieses deshalb, weil alle von Vorgang ausgehenden Beziehungstypen somit ebenfalls vorgangskettenspezifisch abgebildet werden. Eine Identifikation von gleichen Vorgängen innerhalb einer Vorgangsketten als auch vorgangskettenunabhängig kann somit nicht allein aufgrund der Schlüsselattribute erfolgen. Eine derartige Vorgehensweise wäre auch nicht sinnvoll, weil zwei Vorgänge erst dann als gleich einzustufen sind, wenn sowohl die Eingangs- und Ausgangsbeziehungen als auch die den Vorgang triggernden Ereignisse und die aus dem Vorgang resultierenden Ergebnisse gleich sind. Eine Vorgangsgleichheit wird demzufolge innerhalb eines Vorgangskettendiagramms seltener sein als eine Vorgangsähnlichkeit. Vorgangsähnlichkeiten werden durch den Vergleich des Vorgangsnamens, der Input- und Outputbeziehungen sowie der Kontrollstrukturen ermittelt.

### Identifizierende Attribute:

- VKNR (= Vorgangsketten-Identnummer)
- VNR (= Vorgangs-Identnummer)

### Beschreibende Attribute:

- Vorgangname
- klassifizierende Attribut (S -Standard, K-kreativ,...)
- DV-Unterstützungsgrad
- Vorgangsdauer
- mittlere Häufigkeit pro Zeiteinheit der Funktionsausführung
- Wichtigkeit der Funktion
- Voraussetzung für die Vorgangsausführung
- frei definierbare Attribute



## VORGANGSDARSTELLUNG

Eine Vorgangsdarstellung ist eine Zuordnung von einem Vorgangssymbol und einer VKD-Spalte zu einem Vorgang.

### Identifizierende Attribute:

- VNR (= Vorgangs-Identnummer)
- VKNR (= Vorgangsketten-Identnummer)
- VSNR (= Vorgangssymbol-Identnummer)
- VKDNR (=VKD-Identnummer)
- VKDSNR (= VKD-Spalten-Identnummer)

### Beschreibende Attribute:

- X-Koordinate
- Y-Koordinate
- Flächenattribute

## VORGANGS-STRUKTUR

Die Struktur gibt die zeitliche Reihenfolge von Vorgängen an. Weil die Struktur eine Beziehung der Ausprägungen von Vorgängen untereinander darstellt, ist sie ebenfalls wie die Vorgänge selbst vorgangskettenspezifisch. Weil darüber hinaus davon ausgegangen werden kann, daß eine Vorgänger-Nachfolger-Struktur nur zwischen Vorgängen innerhalb einer Vorgangskette dargestellt werden, wird zur Identifikation der Strukturbeziehung das Attribut Vorgangsketten-Identnummer nur einmalig geführt. Wird eine Vorgänger-Nachfolger-Beziehung zwischen zu unterschiedlichen Vorgangsketten gehörigen Vorgängen identifiziert, so sind die einzelnen Vorgangsketten zu einer zusammenzufassen. Die in der Vorgänger-Nachfolgerbeziehung zwischen Vorgängen dargestellte Struktur stellt die Voraussetzung zur Ableitung eines Ablaufmodells dar.

### Identifizierende Attribute:

- VKNR (= Vorgangsketten-Identnummer)
- VVNR (= Vorgängervorgangs-Identnummer)
- NVNR (= Nachfolgervorgangs-Identnummer)

### Beschreibende Attribute:

- /

## VORGANGSSYMBOL (V-SYMBOL)

Das Vorgangssymbol ist die grafische Darstellung eines Vorgangs im VKD.

Das Symbol für die Bearbeitung ist ein Kreis, in dessen Mitte die Vorgangs-Identnummer abgebildet ist, das der Entscheidung ist eine Raute ebenfalls mit einer Vorgangs-Identnummer versehen. Eine Übertragung wird durch einen Pfeil dargestellt, in dessen Mitte ein die Übertragung spezifizierender Kennbuchstabe steht. So steht beispielsweise "T" für Telefon oder "EDI" für Electronic Data Interchange.

Darüber hinaus kann der Vorgang durch weitere klassifizierende Attribute beschrieben werden, wie beispielsweise operativer, standardisierter Vorgang etc.. In Abhängigkeit des Wertes der Ausprägung des Attributes kann eine spezifizierte (und/ oder klassifizierte) Vorgangssymboldarstellung verwendet werden.

### Identifizierende Attribute:

- VSNR (= Vorgangssymbol-Identnummer)

### Beschreibende Attribute:

- Bezeichnung (Bearbeitung; Entscheidung; Übertragung)
- Flächenattribute
- Klassifizierungskürzel (operativ, kreativ,..)
- Spezifizierungskürzel (nur bei Übertragung)

## VORGANGSKETTE (VK)

Eine Vorgangskette ist eine inhaltlich logische, sequentielle Abfolge von Vorgängen, die nicht unterbrochen sein darf. Eine Vorgangskette wird durch ein Ereignis getriggert und mit einem Ergebnisereignis, das keine weitere Vorgänge triggert, abgeschlossen.

Verzweigungen sind aufgrund von Entscheidungen möglich. Eine Vorgangskette wird in mindestens einem VKD grafisch abgebildet.

### Identifizierende Attribute:

- VKNR (= Vorgangsketten-Identnummer)

### Beschreibende Attribute:

- Bezeichnung der Vorgangskette
- Anzahl der enthaltenen Vorgänge
- Modellierungsdatum
- Interviewdatum
- Reviewdatum
- Projekt

## VORGANGSKETTENDIAGRAMM (VKD)

In einem VKD werden Vorgangsketten grafisch und strukturiert abgebildet.

### Identifizierende Attribute:

- VKDNR (= VKD-Identnummer [enthält die Seitenbezeichnung])

### Beschreibende Attribute:

- Name des Vorgangskettendiagramms
- Ersteller
- Erstelldatum
- Projekt
- Versionsnummer
- frei definierbare Attribute
- Darstellungsgröße (*optional: DIN A4*)

## VORGANGSKETTENDIAGRAMMSPALTE (VKD-SPALTE)

Eine Vorgangskettendiagramm-Spalte ist ein Feld eines VKD, in dem Inhalte von Vorgangsketten symbolisch oder in Beschreibungsformen abgebildet werden können.

### Identifizierende Attribute:

- VKDNR (= VKD-Identnummer)
- VKDSNR (= VKD-Spalten-Identnummer)

### Beschreibende Attribute:

- Bezeichnung (Vorgänge, Bearbeitung...)
- Breite (Pixel)
- Anzahl enthaltener Symbole

## ZUORDNUNG Datenspeicher-Vorgangskette

Die Zuordnung zwischen dem Datenspeicher und der Vorgangskette dient zur vorgangskettenspezifischen Analyse von verwendeten Datenspeichermedien.

### Identifizierende Attribute:

- DSNR (= Datenspeicher-Identnummer)
- VKNR (= VK-Identnummer)

### Beschreibende Attribute:

- /

## ZUORDNUNG Datenspeicher-Software

Die Zuordnung weist den Softwaresystemen die zugehörigen Datenspeicher zu.

### Identifizierende Attribute:

- DSNR (= Datenspeicher-Identnummer)
- SWNR (= Software-Identnummer)

### Beschreibende Attribute:

- /



## ZUORDNUNG Vorgang-Funktion

Die Zuordnung weist Vorgängen des VKD Funktionen eines Funktionsmodells zu.

### Identifizierende Attribute:

- VNR (= Vorgangs-Identnummer)
- VKNR (= VK-Identnummer)
- Funktionsbezeichnung
- Funktionsmodellbezeichnung

### Beschreibende Attribute:

- /

### 3.4 Darstellung und Einteilung der im VKD verwendeten Symbole

Die in einem Vorgangskettendiagramm dargestellten Inhalte werden durch Symbole grafisch repräsentiert. Die Symbole werden im beschriebenen Metamodell durch Beziehungstypen zwischen den Inhalten und der VKD-Spalte modelliert. Eine Generalisierung der Symbole ist derzeit nicht vorgesehen, weil alle signifikanten Attribute jeweils den Spezialisierungen zugewiesen werden würden. Dennoch soll an dieser Stelle eine Übersicht über alle im VKD darstellbaren Symbole gegeben werden.

Entsprechend des Metamodells gibt es folgende Arten von Symbolen:

- Vorgangssymbole;
- Datenspeichersymbole;
- Konnektorsymbole;
- Flußsymbole.

Die Symbole werden zusätzlich um Beschreibungen, die ebenfalls im VKD abgebildet werden können, ergänzt. Es existieren:

- Datenbasisbeschreibungen;
- Ressourcenbeschreibungen;
- Vorgangsbeschreibungen.
- Ereignisbeschreibungen;

Die folgende Übersicht ordnet den aufgeführten Symbolen, die als Ausprägungen von Beziehungstypen im Metamodell existieren, grafische Symboldarstellungen zu.

#### **Funktionssymbole:**

- Bearbeitung
- Entscheidung
- Übertragung

#### **Flußdarstellungen**

- Informationsfluß
- Kontrollfluß
- Quelle/Senke-Darstellung

### **Datenspeichersymbole:**

- Datei auf Diskette
- Datei auf Magnetbandspeicher
- Datei auf Mikrofiche
- Datei auf optischer Platte
- Datei auf Plattenspeicher
- DV-Liste
- Kartei
- Manuelle Unterlagen

### **Konnektorsymbole:**

- Konnektor (Quelle)
- Konnektor (Senke)

Die Spezifizierung des Inhalts von Symbolen erfolgt im wesentlichen durch die Attributierung der entsprechenden Informationsobjekte.

An dieser Stelle sollen die einzelnen Symbole sowie ihre Bedeutung dargestellt werden. Die Behandlung der einzelnen Symbole erfolgt in der durch die Klassifizierung vorgegeben Reihenfolge.

### **Funktionssymbole**

Funktionen lassen sich in zwei Spezialisierungen aufteilen:

- Bearbeitung
- und Entscheidung.

Eine Bearbeitung wird durch das in Bild 8 dargestellte Symbol repräsentiert.

Eine Übertragung von Informationen (beispielsweise per Telefon, Fax , etc wird durch ein Sondersymbol der Bearbeitung dargestellt. Dieses ist in Bild 8a dargestellt. Es kann durch einen Kennbuchstaben spezifiziert werden. Es bedeuten:

"T"      Telefon

- "TF" TeleFax
- "TX" Telex
- "P" 'gelbe Post'
- "EDI" Electronic Data Interchange

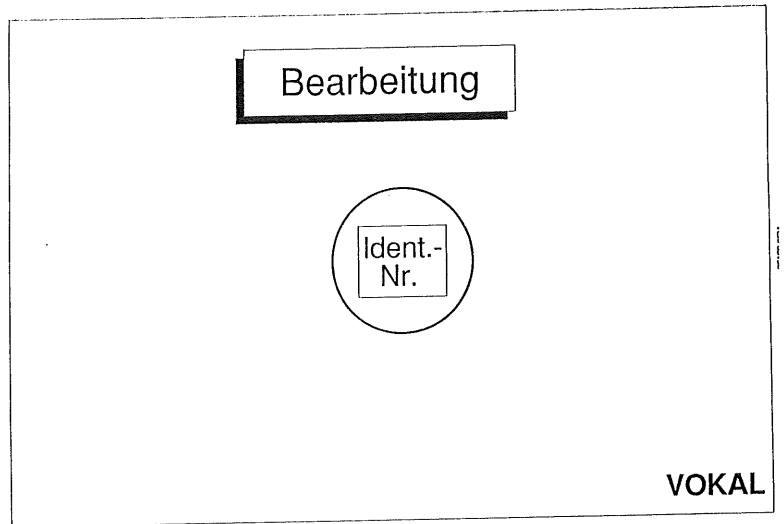


Bild 8: Symbol für die Bearbeitung

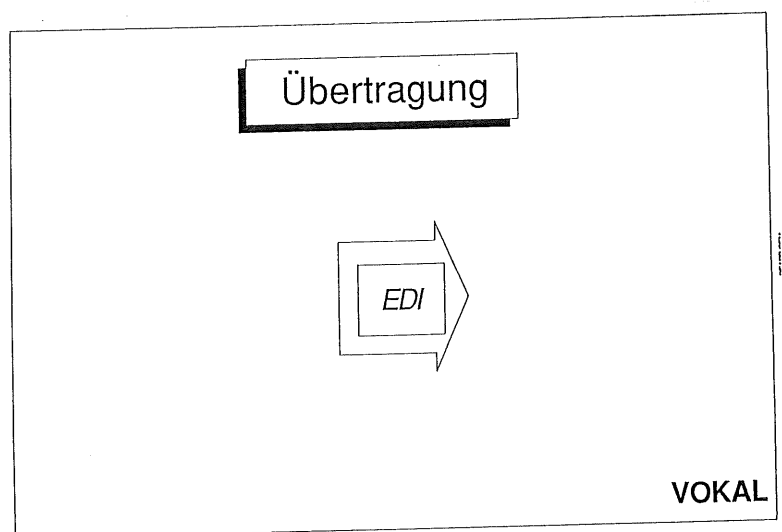


Bild 8a: Symbol der Übertragung

Eine Entscheidung stellt eine Sonderform der Bearbeitung dar. Sie wird durch das in Bild 9 gezeigte Symbol dargestellt.

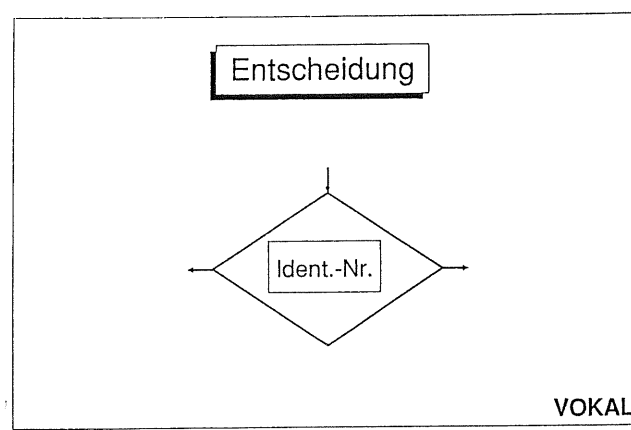


Bild 9: Symbol für die Entscheidung

### Flußdarstellung

Die unterschiedlichen Flußdarstellungen gehen aus dem Bild 10 hervor. Es sind gezeigt der Informationsfluß, der Kontrollfluß und die Quelle/Senke-Darstellung.

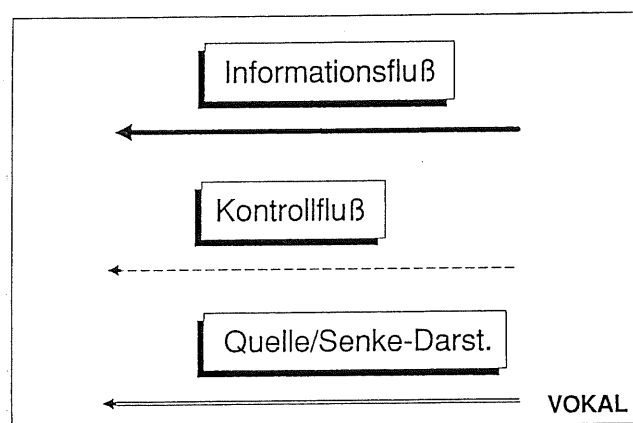


Bild 10: Symbole für Flußdarstellungen

### Datenspeichersymbol

Ein Datenspeicher ist ein physikalisches Objekt, das in der Lage ist, Informationen über die Dauer eines diskreten Zeitintervalls zu speichern. Von der Art des Datenspeichers hängt wesentlich die Weiterverarbeitungsmöglichkeit von Dateninhalten ab. Der Wechsel von Datenspeichern innerhalb einer Vorgangskette kann beispielsweise mit zeitaufwendigen und fehleranfälligen Datenübertragungen verbunden sein. Deshalb dient die Darstellung der Datenspeicher als substantieller Hinweis bei der Analyse von Vorgangsketten. Es ist angezeigt, die Vielfalt der real existierenden Datenspeicher entsprechend des realen Umfangs im Rahmen der Modellierung von Vorgängen abzubilden.

Weil Speichermedien technischen Änderungen unterworfen sind, wird eine vollständige und abgeschlossene

Symboldefinition nicht angestrebt. Im folgenden sind die wichtigsten Datenspeicher in den Bildern 11 bis 18 aufgeführt. Mögliche Erweiterungen hierzu werden durch das freie Definieren von Symbolen ermöglicht.

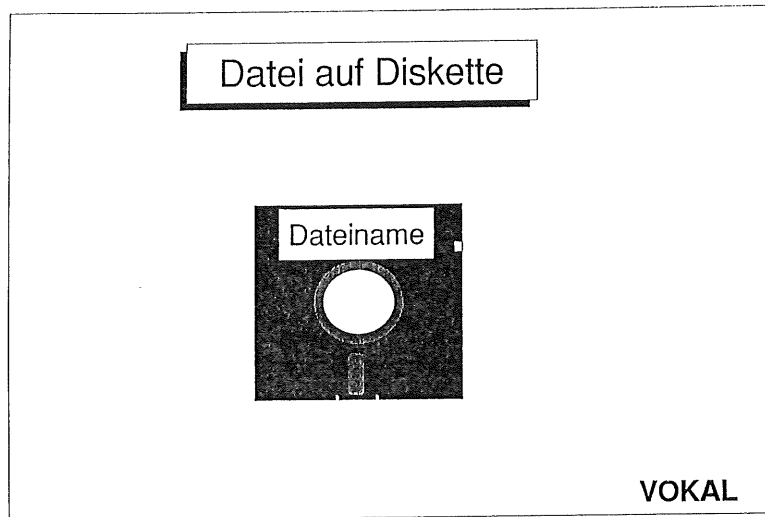


Bild 11: Symbol für eine Datei auf Diskette

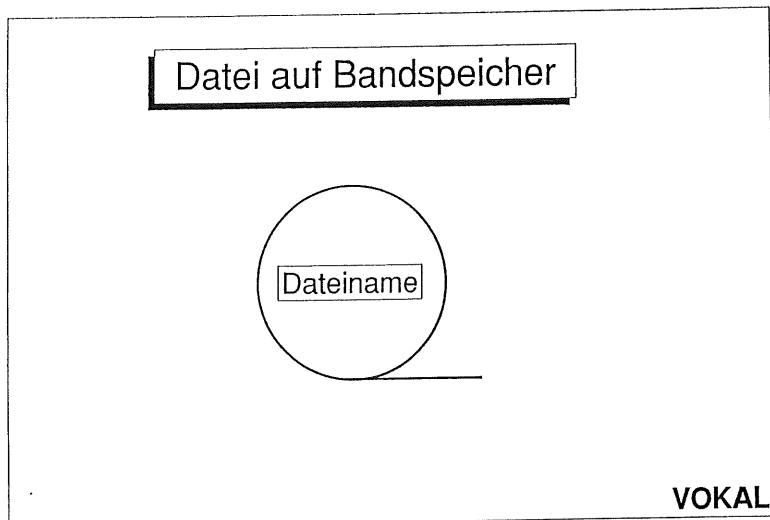


Bild 12: Symbol für eine Datei auf Magnetbandspeicher

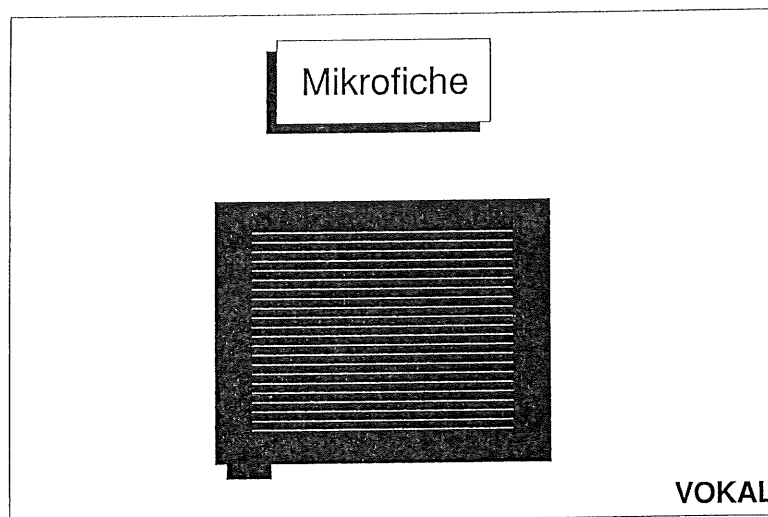


Bild 13: Symbol für eine Datei auf Mikrofiche

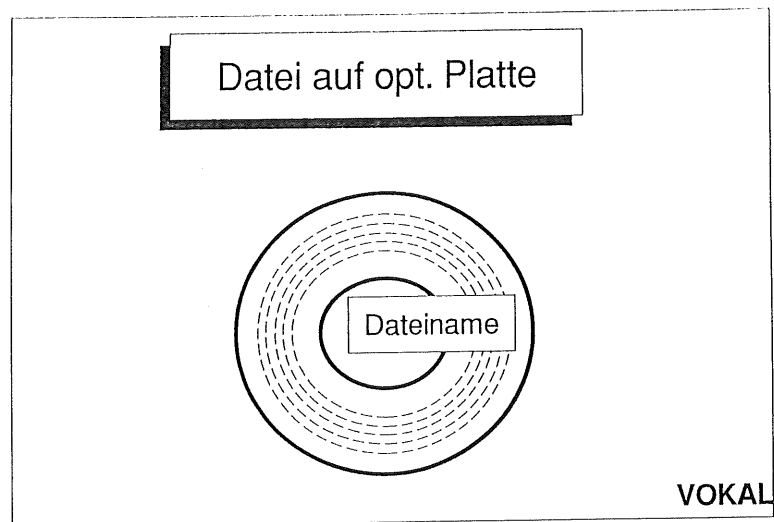


Bild 14: Symbol für eine Datei auf optischer Platte

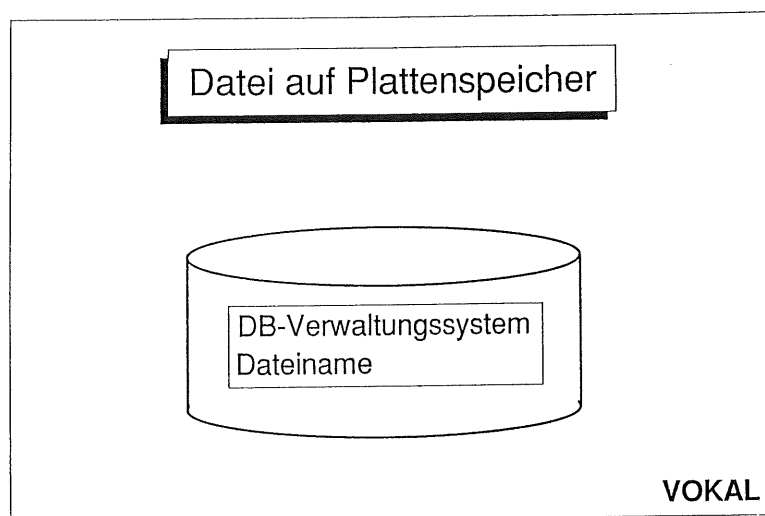


Bild 15: Symbol für eine Datei auf Plattenspeicher

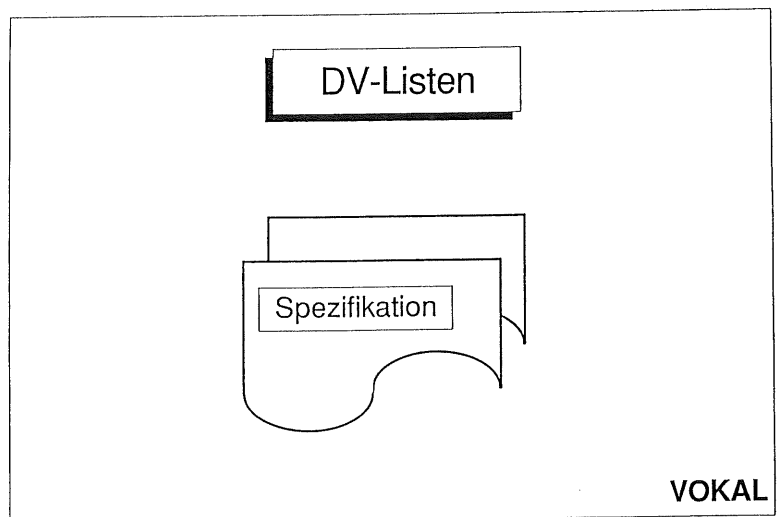


Bild 16: Symbol für eine DV-Liste

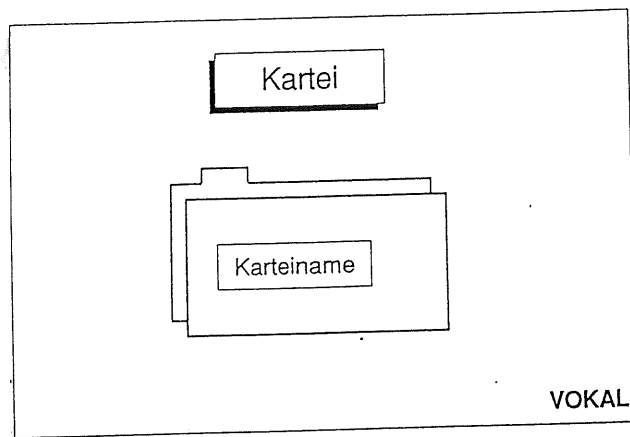


Bild 17: Symbol für eine Kartei

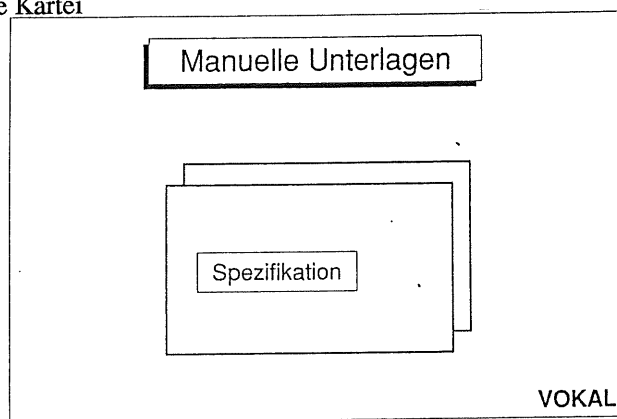


Bild 18: Symbol für manuelle Unterlagen

### Konnektorsymbol

Schnittstellen zwischen VKD sind Konnektoren, die durch die Modellierung von Vorgangsketten entstehen, die sich nicht mit nur einem VKD abbilden lassen. Die Schnittstellen entstehen aber allein durch die Modellierung und sind als inhaltliche Schnittstellen nicht vorhanden.

Das Bild 19 zeigt exemplarisch eine Struktur von über Konnektoren miteinander verbundenen VKD.

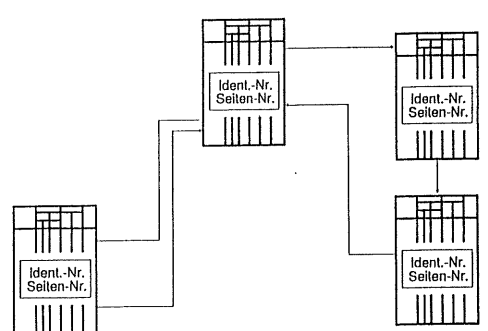


Bild 19: Vorgangskette in mehreren VKD

Das Bild 20 stellt die Symbole für Konnektoren dar.



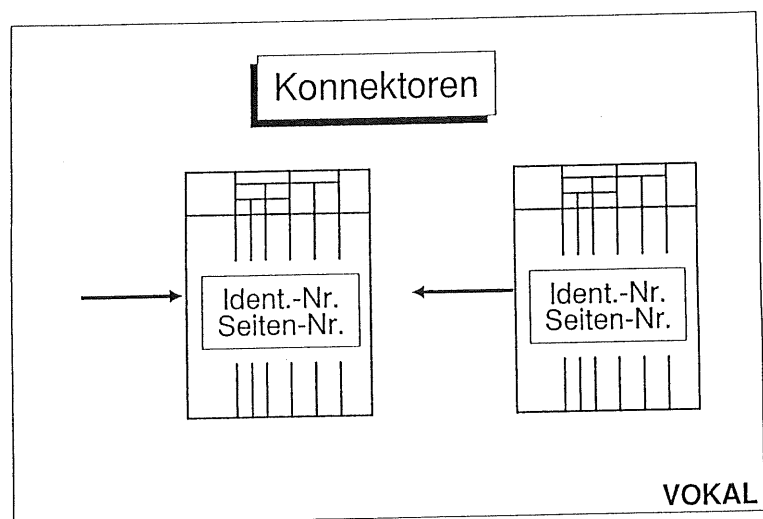


Bild 20: Symbol für einen Konnektor

### 3.5 Verknüpfungsregeln

Der Übersichtlichkeit halber wird erst an dieser Stelle in Ergänzung zu den Darstellungen im Metamodell eine Symbol/Beschreibungsverknüpfungsmatrix dargestellt. Dabei wird angegeben, welche im VKD darstellbaren Symbole in welcher Form miteinander verknüpft werden können.

In der Verknüpfungsmatrix wird auf Ausprägungsebene spezifiziert, welche Symbol- und Beschreibungsverknüpfungen im VKD zulässig sind. Direkt miteinander verbunden werden können nur Symbole, die in der gleichen VKD-Spalte modelliert werden (Vgl. Bild 22).

Die im Bild 21 dargestellte Verknüpfungsmatrix beschreibt die konkreten Verknüpfungsregeln.

In Bild 21 wird durch ein "X" gekennzeichnet, welche Symbole zeichentechnisch direkt miteinander verbunden werden und durch ein "Z", welche Symbole einander zugeordnet werden, indem sie im VKD auf gleicher Höhe abgebildet werden. Ein "I" steht für die Verknüpfung über eine Identnummer.

Die Verknüpfung von Informationsflußdarstellungen, Kontrollflüssen und Quelle/Senke-Darstellungen untereinander erfolgt zur Reduktion der Komplexität von Vorgangsketten nach den bei elektrischen Schaltplänen üblichen Regeln. Dabei werden sich kreuzende Linien ohne Verbindungen übereinander, sich kreuzende Linien mit Verbindungen am Kreuzungspunkt durch einen Punkt kenntlich gemacht. Es können keine Linien mit entgegengesetzter Richtung miteinander verbunden werden, ebenso existieren keine Doppelpfeile. Wie aus der Verknüpfungsmatrix zu entnehmen sind nur gleichartige Pfeildarstellungen also Informationsflüsse untereinander, Kontrollflüsse untereinander und Quelle/Senke-Darstellungen untereinander miteinander verknüpfbar.

Auf Basis dieser Verknüpfungsmatrix können die Regeln zur Plausibilitätsprüfung von Symbol- bzw. Beschreibungseinträgen in das VKD entwickelt werden.

Die Zuordnungen der einzelnen Darstellungselemente zu den VKD-Spalten kann in dem Metamodell auf der Typebene ebenfalls nicht dargestellt werden. Sie sind auf der Ausprägungsebene relevant. Sie sind in Ergänzung zum Metamodell in Bild 22 explizit in der Form einer Symbol-Spalten-Matrix dargestellt:

Bild 21: Symbolverknüpfungsmatrix

Symbol	Datenbasis	Datenspeicher	- D. Bandsp.	- D. Diskette	- D. Mikrofiche	- D. opt. Platte	- D. Plattensp	- DV-Liste	- Kartei	- Man. Unterl	Ereignissymb.	Info-fluß	Konnektorsymb.	Kontrollfluß	MiniSpec	Quelle/Senke-D.	Ressource	Vorgangssymb.
Datenbasis		X																
Datenspeicher	X		X	X	X	X	X	X	X	X		X						
- D. Bandsp.	X											X				X		
- D. Diskette	X											X				X		
- D. Mikrofiche	X											X				X		
- D. opt. Platte	X											X				X		
- D. Plattensp	X											X				X		
- DV-Liste	X											X				X		
- Kartei	X											X				X		
- Man. Unterl	X											X				X		
Ereignissymb.														X				
Info-fluß		X	X	X	X	X	X	X	X	X		X	X					X
Konnektorsymb.													X					
Kontrollfluß										X				X				
MiniSpec																		
Quelle/Senke-D.																X		
Ressource																		
Vorgangssymb.												X			Z,1	X		Z,1

SYMBOL	Vorgang	Datenbasis		Bearbeitung		Ereignis	Ressource
		manuell	DV-unterstützt	DV-unterstützt	manuell		
DATENBASIS		<input type="checkbox"/>	<input type="checkbox"/>				
DATENSPEICHER		<input type="checkbox"/>	<input type="checkbox"/>				
- D. BANDSPEICHER		<input type="checkbox"/>	<input type="checkbox"/>				
- D. DISKETTE		<input type="checkbox"/>	<input type="checkbox"/>				
- D. MIKROFICHE		<input type="checkbox"/>	<input type="checkbox"/>				
- D. OPT. PLATTE		<input type="checkbox"/>	<input type="checkbox"/>				
- D. PLATTENSP.		<input type="checkbox"/>	<input type="checkbox"/>				
- DV-Liste		<input type="checkbox"/>	<input type="checkbox"/>				
- Kartei		<input type="checkbox"/>	<input type="checkbox"/>				
- MANUELLE U (allg.)		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	
EREIGNISSYMBOL						<input type="checkbox"/>	
INFO-FLUSS		<input type="checkbox"/>	<input type="checkbox"/>				<input type="checkbox"/>
KONNEKTORSYMB.						<input type="checkbox"/>	
KONTROLLFLUSS						<input type="checkbox"/>	
MINISPEZIFIKATION	<input type="checkbox"/>						<input type="checkbox"/>
QUELLE/SENKE-D.		<input type="checkbox"/>	<input type="checkbox"/>				<input type="checkbox"/>
RESSOURCE							<input type="checkbox"/>
VORGANGSSYMBOL							<input type="checkbox"/>
- BEARBEITUNG							<input type="checkbox"/>
- ENTSCHEIDUNG							<input type="checkbox"/>

Bild 22: Symbol-Spalten-Matrix des VKD

---

## Literatur

**Grochla, E.:**

Grundlagen der organisatorischen Gestaltung, Stuttgart 1986.

**Jost, W, Keller, G.; Scheer, A.-W.:**

Konzeption eines Tools im Rahmen der CIM-Planung - Gestaltung unternehmensspezifischer Funktions- und Informationsarchitekturen, in: ZfB 61(1991)1, S. 33-64.

**Panse, R.:**

CIM-OSA - Ein herstellerunabhängiges CIM-Konzept, in: DIN-Mitt. 69 (1990) 3, S. 157-164.

**Scheer, A.-W.:**

Architektur integrierter Informationssysteme - Grundlagen der Informationsmodellierung, Berlin et al. 1991.

**Scheer, A.-W.:**

CIM - Der computergesteuerte Industriebetrieb, 4. Aufl., Berlin et al. 1990a.

**Scheer, A.-W.:**

EDV-orientierte Betriebswirtschaftslehre - Grundlagen für ein effizientes Informationsmanagement, 4. Aufl., Berlin et al. 1990b

**Scheer, A.-W.:**

Wirtschaftsinformatik - Informationssysteme im Industriebetrieb, 3. Aufl., Berlin et al. 1990c.

**Scheer, A.-W.:**

Modellierung betriebswirtschaftlicher Informationssysteme (Teil 1: Logisches Informationsmodell) in: Scheer,

A.-W. (Hrsg.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 67, Saarbrücken 1990d.

**Scheer, A.-W.; Berkau, C.; Kruse, Chr.:**

Analyse der Umsetzung einer EDI-Konzeption am Beispiel der Beschaffungslogistik in der Automobilzulieferindustrie, in: IM 5(1991)3, S. 30-37.

**Scheer, A.-W.; Berkau, C.; Kruse; Chr.; Rehaag, E.:**

Ablauforganisatorisches Rationalisierungspotential durch EDI in der Auftragsabwicklung, in Automobilindustrie 36 (1991)5, Veröffentlichung in Druck.

**Scheer, A.-W.; Kraemer, W.:**

Betriebsübergreifende Vorgangsketten und Informationssysteme, in: CIM Management 5 (1989) 3, S. 4-9.

**Schlageter, G.; Stucky, W.:**

Datenbanksysteme: Konzepte, und Modelle, 2. Aufl., Stuttgart 1983.

**Stotko, E.:**

CIM-OSA - Europäische Initiative für offene CIM-Systeme - Architektur, in: CIM Management 5 (1989) 1, S. 9-15.

Die Veröffentlichungen des Instituts für Wirtschaftsinformatik (IWi) im Institut für empirische Wirtschaftsforschung an der Universität des Saarlandes erscheinen in unregelmäßiger Folge.

\* Die Hefte 1 - 31 werden nicht mehr verlegt.

- Heft 32: A.-W. Scheer: Einfluß neuer Informationstechnologien auf Methoden und Konzepte der Unternehmensplanung, März 1982, Vortrag anläßlich des Anwendergespräches "Unternehmensplanung und Steuerung in den 80er Jahren in Hamburg vom 24. - 25.11.1981
- Heft 33: A.-W. Scheer: Dispositio- und Bestellwesen als Baustein zu integrierten Warenwirtschaftssystemen, März 1982, Vortrag anläßlich des gdi-Seminars "Integrierte Warenwirtschafts-Systeme" in Zürich vom 10. - 12. Dezember 1981
- Heft 34: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS - Ein Ansatz zur Entwicklung prüfungsgerechter Software-Systeme, Mai 1982
- Heft 35: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Konzept einer computergestützten Prüfungsumgebung, Juli 1982
- Heft 36: A.-W. Scheer: Rationalisierungserfolge durch Einsatz der EDV - Ziel und Wirklichkeit, August 1982, Vortrag anläßlich der 3. Saarbrücker Arbeitstagung "Rationalisierung" in Saarbrücken vom 04. - 06. 10.1982
- Heft 37: A.-W. Scheer: DV-gestützte Planungs- und Informationssysteme im Produktionsbereich, September 1982
- Heft 38: A.-W. Scheer: Interaktive Methodenbanken: Benutzerfreundliche Datenanalyse in der Marktforschung, Mai 1983
- Heft 39: A.-W. Scheer: Personal Computing - EDV-Einsatz in Fachabteilungen, Juni 1983
- Heft 40: A.-W. Scheer: Strategische Entscheidungen bei der Gestaltung EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 41: H. Krcmar: Schnittstellenprobleme EDV-gestützter Systeme des Rechnungswesens, August 1983, Vortrag anläßlich der 4. Saarbrücker Arbeitstagung "Rechnungswesen und EDV" in Saarbrücken vom 26. - 28.09.1983
- Heft 42: A.-W. Scheer: Factory of the Future, Vorträge im Fachausschuß "Informatik in Produktion und Materialwirtschaft" der Gesellschaft für Informatik e. V., Dezember 1983
- Heft 43: A.-W. Scheer: Einführungsstrategie für ein betriebliches Personal-Computer-Konzept, März 1984
- Heft 44: A.-W. Scheer: Schnittstellen zwischen betriebswirtschaftlicher und technische Datenverarbeitung in der Fabrik der Zukunft, Juli 1984
- Heft 45: J. Ahlers, W. Emmerich, H. Krcmar, A. Pocsay, A.-W. Scheer, D. Siebert: EPSOS-D, Ein Werkzeug zur Messung der Qualität von Software-Systemen, August 1984
- Heft 46: H. Krcmar: Die Gestaltung von Computer am-Arbeitsplatz-Systemen - ablauforientierte Planung durch Simulation, August 1984
- Heft 47: A.-W. Scheer: Integration des Personal Computers in EDV-Systeme zur Kostenrechnung, August 1984

- Heft 48: A.-W. Scheer: Kriterien für die Aufgabenverteilung in Mikro-Mainframe Anwendungssystemen, April 1985
- Heft 49: A.-W. Scheer: Wirtschaftlichkeitsfaktoren EDV-orientierter betriebswirtschaftlicher Problemlösungen, Juni 1985
- Heft 50: A.-W. Scheer: Konstruktionsbegleitende Kalkulation in CIM-Systemen, August 1985
- Heft 51: A.-W. Scheer: Strategie zur Entwicklung eines CIM-Konzeptes - Organisatorische Entscheidungen bei der CIM-Implementierung, Mai 1986
- Heft 52: P. Loos, T. Ruffing: Verteilte Produktionsplanung und -steuerung unter Einsatz von Mikrocomputern, Juni 1986
- Heft 53: A.-W. Scheer: Neue Architektur für EDV-Systeme zur Produktionsplanung und -steuerung, Juli 1986
- Heft 54: U. Leismann, E. Sick: Konzeption eines Bildschirmtext-gestützten Warenwirtschaftssystems zur Kommunikation in verzweigten Handelsunternehmungen, August 1986
- Heft 55: D. Steinmann: Expertensysteme (ES) in der Produktionsplanung und -steuerung (PPS) unter CIM-Aspekten, November 1987, Vortrag anlässlich der Fachtagung "Expertensysteme in der Produktion" am 16. und 17.11.1987 in München
- Heft 56: A.-W. Scheer: Enterprise wide Data Model (EDM) as a Basis for Integrated Information Systems, Juli 1988
- Heft 57: A.-W. Scheer: Present Trends of the CIM Implementation (A qualitative Survey) Juli 1988
- Heft 58: A.-W. Scheer: CIM in den USA - Stand der Forschung, Entwicklung und Anwendung, November 1988
- Heft 59: R. Herterich, M. Zell: Interaktive Fertigungssteuerung teilautonomer Bereiche, November 1988
- Heft 60: A.-W. Scheer, W. Kraemer: Konzeption und Realisierung eines Expertenunterstützungssystems im Controlling, Januar 1989
- Heft 61: A.-W. Scheer, G. Keller, R. Bartels: Organisatorische Konsequenzen des Einsatzes von Computer Aided Design (CAD) im Rahmen von CIM, Januar 1989
- Heft 62: M. Zell, A.-W. Scheer: Simulation als Entscheidungsunterstützungsinstrument in CIM, September 1989
- Heft 63: A.-W. Scheer: Unternehmens-Datenbanken - Der Weg zu bereichsübergreifenden Datenstrukturen, September 1989
- Heft 64: C. Berkau, W. Kraemer, A.-W. Scheer: Strategische CIM-Konzeption durch Eigenentwicklung von CIM-Modulen und Einsatz von Standardsoftware, Dezember 1989
- Heft 65: A. Hars, A.-W. Scheer: Entwicklungsstand von Leitständen<sup>[1]</sup>, Dezember 1989



- Heft 66: W. Jost, G. Keller, A.-W. Scheer: CIMAN - Konzeption eines DV-Tools zur Gestaltung einer CIM-orientierten Unternehmensarchitektur, März 1990
- Heft 67: A.-W. Scheer: Modellierung betriebswirtschaftlicher Informationssysteme (Teil 1: Logisches Informationsmodell), März 1990
- Heft 68: W. Kraemer: Einsatzmöglichkeiten von Expertensystemen in betriebswirtschaftlichen Anwendungsgebieten, März 1990
- Heft 69: A.-W. Scheer, R. Bartels, G. Keller: Konzeption zur personalorientierten CIM-Einführung, April 1990
- Heft 70: St. Spang, K. Ibach: Zum Entwicklungsstand von Marketing-Informationssystemen in der Bundesrepublik Deutschland, September 1990
- Heft 71: D. Aue, M. Baresch, G. Keller: URMELE, Ein Unternehmensmodellierungsansatz, Oktober 1990
- Heft 72: M. Zell: Datenmanagement simulationsgestützter Entscheidungsprozesse am Beispiel der Fertigungssteuerung, November 1990
- Heft 73: A.-W. Scheer, M. Bock, R. Bock: Expertensystem zur konstruktionsbegleitenden Kalkulation, November 1990
- Heft 74: R. Bartels, A.-W. Scheer: Ein Gruppenkonzept zur CIM-Einführung, Januar 1991
- Heft 75: M. Nüttgens, St. Eichacker, A.-W. Scheer: CIM-Qualifizierungskonzept für Klein- und Mittelunternehmen (KMU), Januar 1991
- Heft 76: Ch. Houy, J. Klein: Die Vernetzungsstrategie des Instituts für Wirtschaftsinformatik - Migration vom PC-Netzwerk zum Wide Area Network (noch nicht veröffentlicht)
- Heft 77: W. Kraemer: Ausgewählte Aspekte zum Stand der EDV-Unterstützung für das Kostenmanagement: Modellierung benutzerindividueller Auswertungssichten in einem wissensbasierten Controlling-Leitstand, Mai 1991
- Heft 78: H. Heß: Vergleich von Methoden zum objektorientierten Design von Softwaresystemen, August 1991
- Heft 79: A.-W. Scheer: Konsequenzen für die Betriebswirtschaftslehre aus der Entwicklung der Informations- und Kommunikationstechnologien, Mai 1991
- Heft 80: G. Keller, J. Kirsch, M. Nüttgens, A.-W. Scheer: Informationsmodellierung in der Fertigungssteuerung, August 1991
- Heft 81: A.-W. Scheer: Papierlose Beratung - Werkzeugunterstützung bei der DV-Beratung, August 1991
- Heft 82: C. Berkau: VOKAL (System zur Vorgangskettendarstellung und -analyse) - Struktur der Modellierungsmethode - Juni 1991 (wird nicht verlegt)