

Das Interaktive Daten-Anwendungssystem

Objektorientiertes Arbeiten zur Definition, Entwicklung und Integration von DV-Anwendungssystemen

Von Miklós Géza Zilahi-Szabó, Andreas van de Valk und Gerd Koslowski

Der Name IDAS wird aus wettbewerbsrechtlichen Gründen geschützt

In den letzten Jahren ist der Personal-Computer-Einsatz in Unternehmen Realität geworden. Die zunehmende Vielfalt verfügbarer Software läßt Integrationswünsche in den Vordergrund treten, die vielfach durch die derzeit erhältlichen Produkte und angewandten Entwicklungsmethoden und -techniken nicht erfüllt werden können. Bislang wurden DV-Systeme (Computerprogramme) für eine bestimmte Aufgabenstellung dahingehend konzipiert, daß der Anwender seine Daten nur in einem „vorgefertigten“ Rahmen und in einer „vorgeschriebenen“ Abfolge verarbeiten konnte. Diese Systeme passen sich nicht den Bedürfnissen des Anwenders an. Im Gegenteil, der Anwender muß das DV-System „bedienen“, d.h. er

muß sich die Abfolge erarbeiten und kennenlernen, um das von ihm gewünschte Ergebnis fehlerfrei zu erhalten. Dieser Sachverhalt ist nur solange tragbar, wie die technischen Möglichkeiten unzureichend sind. Durch verbesserte Speicherungsmöglichkeiten der Daten in „Datenbanken“, neue DV-Techniken sowie theoretische Vorarbeiten (Anwenderbedürfnisse und Wissensspeicherung/-verarbeitung) ist der Zeitpunkt gekommen, den umgekehrten Weg zu gehen. Das DV-System „bedient“ den Anwender, in dem es sich seinen Bedürfnissen anpaßt und das notwendige Wissen bereithält (aber auch vom Anwender „hinzulernen“ kann), um die gestellte Aufgabe zu lösen.

Die elektronische Datenverarbeitung befindet sich auf dem Weg zu Anwendungs- oder wissensbasierten Systemen. Bislang wurden DV-Systeme für eine bestimmte Aufgabenstellung dahingehend konzipiert, daß der Anwender seine Daten nur in einem „vorgefertigten“ Rahmen und in einer „vorgeschriebenen“ Abfolge verarbeiten kann. Diese Systeme passen sich nicht den Bedürfnissen des Anwenders an. Im Gegenteil, der Anwender muß das

DV-System „bedienen“, d.h. er muß sich die Abfolge erarbeiten und kennenlernen, um das von ihm gewünschte Ergebnis fehlerfrei zu erhalten.

Für die Softwareentwicklung stellt sich damit die Frage, mit welchen Methoden und Technologien sie integrative Aspekte moderner Technik wirtschaftlich umsetzen kann. Ein Weg dazu ist, die Daten objektorientiert zu modellieren und in den Anwendungs-Systemen zu benutzen. Diesen Weg bestreitet

IDAS, das Interaktive Daten-Anwendungssystem¹ (vgl. Abb. 1).

Datenmodellierung

Viele große Softwaresysteme benötigen große Datenbanken. Während des Programmablaufes lesen diese Systeme Informationen aus den Datenbanken, fügen neue Informationen hinzu oder verändern bereits bestehende Informationen. Entweder existieren diese Datenbanken unabhängig von den Softwaresystemen oder sie werden speziell für ein bestimmtes System erstellt. Eine Aufgabe der Systementwicklung ist es, die logische Form dieser Datenbanken zu entwickeln.

Objektorientierung

Der objektorientierte Ansatz ist eine Methode, neue Softwaresysteme nicht basierend auf Funktionen, die nach außen nicht bekannte Datenobjekte verändern, sondern auf Objekte, welche Daten und Funktionen umfassen, zu entwickeln. Diese Methode maximiert die Verkapselung von Informationen und führt in vielen Fällen zu Systemen mit geringerer Komplexität als der funktionale Ansatz. In unterschiedlichen Objekten können Methoden gleichen Namens mit unterschiedlichen Wirkungen existieren. Dadurch bewirkt z.B.

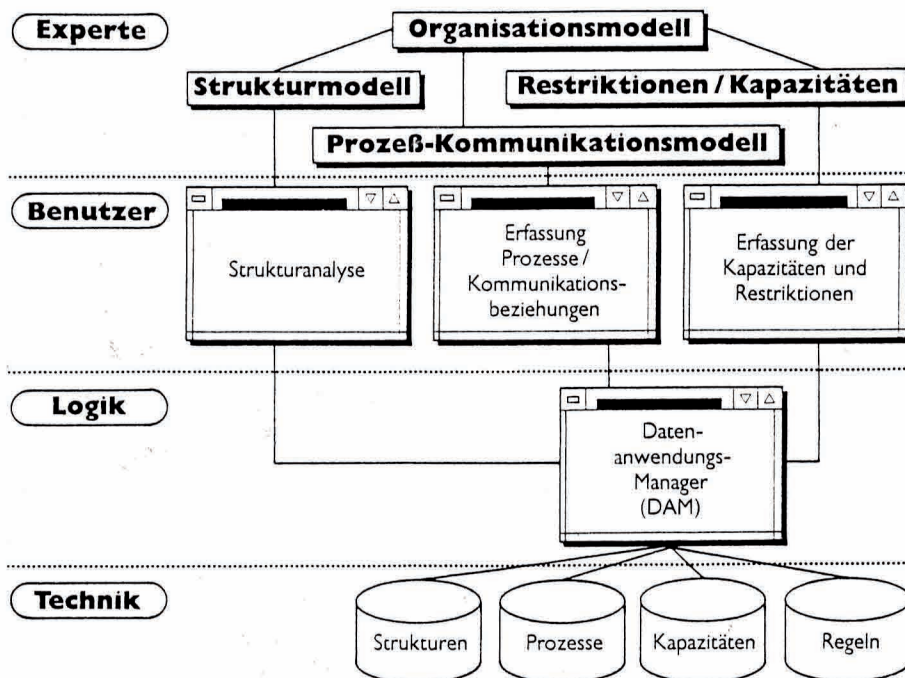


Abb. 1: Das Konzept von IDAS.

¹ IDAS wurde in Kooperation mit der Fa. HEC Hanseatische Software-Entwicklungs- und Consulting GmbH Bremen entwickelt

die Methode „Invertieren“ für ein Objekt vom Typ Skalar die Berechnung seines Kehrwertes, für ein Objekt vom Typ Matrix eine Matrixinversion und für ein Objekt vom Typ Bildelement die Umkehrung seiner Farbattribute von Schwarz und Weiß und umgekehrt.

Kommunikation zwischen Objekten findet durch Austausch von Nachrichten statt. Die Elemente einer Nachricht sind der Name des Empfängers, die Selektoren (Namen von Methoden, welche das Empfängerobjekt ausführen soll) und möglicherweise die Argumente (aktuelle Parameter) für die auszuführenden Methoden.

Jedes Objekt „erbt“ die Eigenschaften seiner Klasse, d.h. die Methoden und die Variablen (Instanzvariablen genannt), welche die in der Klasse vereinbarte Struktur besitzen und die bei der Nachricht zu ihrer Erzeugung mitgegebenen Werte erhalten oder durch eine ihrer (ererbten) Methoden initialisiert werden. Die Klasse ist aus der Sicht der Datenstruktur vergleichbar mit einer Satzdefinition (Record-Deklaration), nur mit dem Unterschied, daß bei der Bildung einer Instanz der Klasse auch die Methoden vererbt werden.

Ein System, welches durchgehend objektorientiert ist, betrachtet Klassen ebenfalls als Objekte, welche wiederum Instanzen einer übergeordneten Klasse (Superklasse) sein können. Dieser Prozeß der Abstraktion bzw. Klassenbildung kann (beliebig) fortgesetzt werden. Es entsteht eine Hierarchie von Klassen, in welcher jede Klasse jeweils alle Eigenschaften ihrer Superklasse erbt.

Benutzeroberfläche

Die Benutzerschnittstelle ist das Mittel, mit dem Menschen und Computer miteinander kommunizieren. Ähnlich wie ein Dialog oder eine Interaktion zwischen Menschen benötigt auch der Dialog Mensch-Maschine Regeln.

- Diese Regeln müssen
- theoretisch begründet,
 - konsistent und
 - vorteilhaft sein.

Die Benutzerschnittstelle besteht dabei aus drei Hauptkomponenten:

- Präsentationssprache,
- Aktionsprache und
- konzeptionellem Benutzermodell.



Das Interaktive Daten-Anwendungssystem IDAS ist ein benutzerfreundliches Programm für Software-Entwicklung.

Foto: Lakos

Die Art, wie die Maschine mit dem Benutzer kommuniziert, wird durch die Computeranwendung bestimmt und als die **Präsentationssprache** bezeichnet. Diese bestimmt, wie auf Informationen zugegriffen wird, Berechnungen durchgeführt werden und wie die ermittelten Informationen in einer verständlichen und benutzbaren Form dem Benutzer präsentiert werden.

Die Benutzer müssen die Information, die ihnen der Computer mittels der **Präsentationssprache** präsentiert, verstehen und mit einer der Information entsprechenden Antwort reagieren. Die Regeln für die Art und Weise der Sendung einer Antwort, bestehend aus Tastatureingaben oder Mausebetätigungen, ist der zweite Teil der Schnittstelle, die **Aktionssprache**.

Der dritte Teil der Schnittstelle entwickelt sich aus dem Verständnis des Benutzers des Computersystems von der Struktur des Kommunikationsprozesses mit der EDV-Anlage. Dieser Teil wird das **konzeptionelle Benutzermodell** genannt. Es setzt sich zusammen aus den Erwartungen des Benutzers bezüglich dessen, was eine Computerschnittstelle darstellt, was sie macht und wie sie funktioniert.

Inhalte der vorgestellten Software

Der heute erreichte Stand der Datenbanktechnologie, die Verfügbarkeit von kostengünstigen Rechnern, die Möglichkeiten der Interprozesskommunikation und die Window-Technologie ermöglichen die Implementierung der seit mehr als 20 Jahren theoretisch entwickelten und praktisch erprobten Konzepte. Ein typisches Beispiel dafür ist das IDAS.

IDAS als integriertes Daten-Anwendungssystem erfüllt als Prototyp bereits wesentliche Anforderungen an ein solches „dienendes DV-System“. Ein Experte definiert mit IDAS in Abstimmung mit dem späteren Benutzer ein Datenmodell. Er legt dabei den Umfang und die Struktur der Daten fest. Ferner werden Restriktionen und Kapazitäten ermittelt. Am Ende entsteht ein Daten- und Prozeß-Kommunikationsmodell, das den Bedürfnissen des Benutzers entspricht. Die Arbeit mit IDAS läuft dabei wie folgt ab: Es werden zuerst Beschreibungselemente („Deskriptoren“) definiert, die zu Klassen („Daten- und Deskriptorenklassen“) zusammengefaßt werden. Aus den Fragestellungen des Benutzers werden dann Möglichkeiten zur wertmäßigen

Erfassung des Datengerüsts generiert. In einem interaktiven Dialog wird der Benutzer angehalten, seine Daten vollständig und konsistent zu erfassen. Diese Daten können dann flexibel ausgewertet und weiteren Anwendungsprogrammen zur Verfügung gestellt werden. Diese Arbeit erledigt der Daten-Anwendungs-Manager von IDAS. Das integrierte Daten-Anwendungssystem ist derzeit auf die Definition, Entwicklung und Integration von DV-Anwendungssystemen in neuzugestaltende und bestehende Organisationsstrukturen beschränkt (Abb. 2).

Aus der Analyse einer Organisation wird ein datenverarbeitender Prozeß isoliert. Dieser Prozeß wird mit seiner internen Datenstruktur, seinen externen Datenschnittstellen und den Regeln für seine Datendynamik definiert und im System abgebildet.

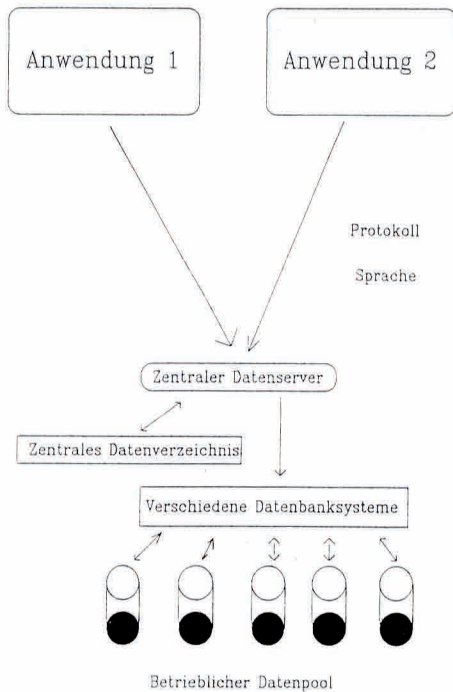


Abb. 2: Daten- und Anwendungsintegration mit zentralem Datenmodellserver.

Abb. 3: Eingabemaske zur Bildung von Datenmodellen.

Ein zentraler Datenmanager(-server) interpretiert die Datenmodellbeschreibung und gibt Anwendungen, die diesen Prozeß DV-technisch unterstützen, einen strukturierten Zugriff auf die Daten. Allgemeingültige, mit diesem Datenmanager kommunizierende Software-Werkzeuge machen eine Individualprogrammierung nur noch für Spezialfälle erforderlich. Diese können durch den Zugriff auf den Datenmanager über Standardschnittstellen besonders kostengünstig entwickelt werden (Abb. 3 und 4).

Arbeiten mit IDAS

Anhand eines Beispiels kann die Arbeit mit IDAS verdeutlicht werden. Dabei werden Vereinfachungen vorgenommen, um die Übersichtlichkeit zu wahren.

Ein EDV-Systemhaus vertreibt Computerprogramme (Software) und die dafür benötigten Personal-Computer und Zusatzgeräte (Hardware). Zwecks Vereinfachung wird unterstellt, daß das EDV-Systemhaus jeweils vier Hard- und Software-Produkte in seinem Produk-

Abb. 4: Eingabemaske zur Bildung von Datenklassen.

tionsprogramm hat. Aufgrund der technologischen Entwicklung sind zwei Produkte der Hardware veraltet und sollen aus dem Produktionsprogramm genommen und durch neue ersetzt werden. Andererseits sind Kapazitäten bei Mitarbeitern in der Programmierung verfügbar, so daß neue Software erstellt werden kann. Außerdem verfügt das Unternehmen über liquide Finanzmittel, die es investieren kann. Der Geschäftsführer hat somit das Problem, sein Produktionsprogramm neu zu gestalten. Es stellen sich ihm mehrere Fragen:

1. Welche Hardware-Produkte sind zu ersetzen?
2. Welche Software-Produkte verlangt der Kundenkreis?
3. Welche Mengen und welche Zeiten sind zu planen?
4. Was bringt es an Einsparungen?
5. Welche Kosten werden wo entstehen? etc.

Der traditionelle Lösungsweg

Für die Beantwortung seiner Fragen müßte der Geschäftsführer in der traditionellen Vorgehensweise die Mitarbeiter aus verschiedenen Bereichen (Marketing, Vertrieb, Produktion, Verwaltung, etc.) ansprechen, um die Information (Daten) zur Beantwortung der obigen Fragen zu erhalten.

Tabelle 1: Daten des Einkaufs

Bezeichnung Komponenten	Einkaufspreis (DM)	Beschaffungsdauer
1. Zentraleinheit 286	500,-	1 Woche
2. Zentraleinheit 386sx	1000,-	2 Wochen
3. Zentraleinheit 386dx	2000,-	2 Wochen
4. Festplatte 40 MB	400,-	4 Tage
5. Festplatte 150 MB	1250,-	1 Woche
6. Festplatte 350 MB	3500,-	1 Woche
7. Festplatte 600 MB	5500,-	1 Woche
8. Monitor einfarbig	150,-	3 Tage
9. Monitor farbig	650,-	2 Tage
10. Laufwerk 360 KB	100,-	3 Tage
11. Laufwerk 720 KB	150,-	4 Tage
12. Laufwerk 1,2 MB	150,-	3 Tage
13. Laufwerk 1,44 MB	200,-	1 Woche
14. Betriebssystem PC	150,-	2 Tage
15. Betriebssystem Netz-1	3000,-	2 Wochen
16. Betriebssystem Netz-2	8000,-	2 Wochen
17. Programmiersprache	1500,-	3 Tage
18. Disketten 5 1/4 (100 St.)	100,-	1 Woche
19. Disketten 3 1/2 (100 St.)	200,-	1 Woche

Diese Informationen muß er dann nach seinen Vorstellungen für die Problemlösung so aufbereiten, daß er sie für Berechnungen verwenden kann. Seine Mitarbeiter liefern ihm die Informationen entweder über manuell geführte Listen und/oder spezielle Computerprogramme für ihre Bereiche. In beiden Fällen sind die Informationen (siehe Tabellen 1-4) für den Geschäftsführer jeweils zu ermitteln und durch diesen neu und damit doppelt zu erfassen.

Tabelle 2: Daten der Produktion

Hardware		
Bezeichnung Produkt	Komponenten	Produktionsdauer
20. Netzwerk-Server-386dx für 4 Arbeitsplätze	3, 5, 8, 12, 13, 14, 15	2 Tage
21. Netzwerk-Server-386dx für 8 Arbeitsplätze	3, 6, 8, 12, 13, 14, 16	2 Tage
22. Arbeitsstation-286 („diskless“)	1, 8, 14	1 Tag
23. Arbeitsstation-286 („intelligent“)	1, 4, 8, 12, 14	1 Tag
Software (Jahreswerte)		
Bezeichnung Produkt	Personalkosten (DM)	Produktionsdauer
24. Finanzbuchhaltung	300 000,-	3 Mannjahre
25. Lohnbuchhaltung	400 000,-	4 Mannjahre
26. Fakturierung	350 000,-	3,5 Mannjahre
27. Dienstprogramme	200 000,-	2 Mannjahre
28. Wartung/Pflege jeweils freie Kapazitäten	150 000,-	1,5 Mannjahre
	200 000,-	2 Mannjahre

Tabelle 3: Daten des Verkaufs

Hardware			
Bezeichnung Produkt	Verkäufe		Verkaufspreis (DM)
	Vorjahr	Jahr	
20. Netzwerk-Server-386dx für 4 Arbeitsplätze	50	70	12 500,-
21. Netzwerk-Server-386dx für 8 Arbeitsplätze	30	90	25 000,-
22. Arbeitsstation-286 („diskless“)	250	170	1 500,-
23. Arbeitsstation-286 („intelligent“)	100	180	2 500,-
Software (Jahreswerte)			
Bezeichnung Produkt	Verkäufe		Verkaufspreis (DM)
	bislang	Jahr	
24. Finanzbuchhaltung	450	35	7 500
25. Lohnbuchhaltung	300	15	9 500
26. Fakturierung	400	40	7 000
27. Dienstprogramme	450	40	3 000
28. Wartung/Pflege jeweils freie Kapazitäten	450	40	900

Tabelle 4: Daten des Vertriebs

Hardware		
Bezeichnung Produkt	Einschätzung	Empfehlung
20. Netzwerk-Server-386dx für 4 Arbeitsplätze	Steigend	Keine Änderung
21. Netzwerk-Server-386dx für 8 Arbeitsplätze	Stark steigend	Keine Änderung
22. Arbeitsstation-286 („diskless“)	Veraltet	Wechsel auf 386sx
23. Arbeitsstation-286 („intelligent“)	Wird stagnieren	Wechsel auf 386sx
Software (Jahreswerte)		
Bezeichnung Produkt	Einschätzung	Empfehlung
24. Finanzbuchhaltung	Zu überarbeiten	Benutzerführung
25. Lohnbuchhaltung	Zu überarbeiten	Baulohn aufnehmen
26. Fakturierung	Deckt alles ab	Keine Änderung
27. Dienstprogramme	Erweitern	Netz-Utilities
28. Wartung/Pflege jeweils freie Kapazitäten	Zu billig	VK-Preis erhöhen
29. Lagerverwaltung	Abrundung	Neues Produkt

Neben diesen Daten der Mitarbeiter verfügt der Geschäftsführer über eigene Informationen, wie z.B. den Personalstand oder die Verkaufspreise der Konkurrenz. Er kann nun mittels manueller Berechnungen oder unter Zuhilfenahme eines Computerprogramms (z.B. einer Tabellenkalkulation) seine Berechnungen vornehmen.

Er beginnt mit der Ermittlung des Deckungsbeitrages für jedes einzelne Produkt des bisherigen Produktionsprogramms und stellt diesen die der neuen Produkte (z.B. „Arbeitsstation-386sx [diskless]“, Schätzwerte für die Erstellung der „Lagerverwaltung“) gegenüber. Nach Abschluß seiner Berechnungen, die immer wieder durch Rückfragen oder Besprechungen mit Mitarbeitern der einzelnen Bereiche begleitet sind, steht die argumentative Vermarktung des neuen Produktionsprogramms. Während die Hardware-Produkte unverzüglich in Produktion gehen können (lediglich die Beschaffungsdauer der Zentraleinheit erhöht sich um eine Woche), kann das Lagerhaltungsprogramm nur mit seinem voraussichtlichen Fertigstellungstermin angekündigt werden.

Der Lösungsweg mit IDAS

Für die Beantwortung seiner Fragen mit IDAS hätte der Geschäftsführer unter Zuhilfenahme

eines IDAS-Experten einen vollkommen anderen Lösungsweg wahrgenommen. Mittels der „Top-Down-Strategie“ (Analyse, Entwurf, Test etc. eines hierarchisch gegliederten Systems von „oben-nach-unten“) werden Unternehmensprozesse als Gruppen logisch zusammenhängender Entscheidungen und Aktivitäten definiert. Nachdem diese vollständig ermittelt und beschrieben wurden, werden sie mit der Organisationsstruktur verknüpft. Es entsteht eine Organisations-/Prozeß-Matrix (siehe Tabelle 5).

Somit ist die Organisation des „EDV-Systemhaus“ beschrieben. In einem zweiten Schritt wird das geplante Anwendungssystem für die Problemstellung fixiert. Im vorliegenden Fall ist dies das Produktionsprogramm. Hierfür wird ein Datenmodell festgelegt. Datenmodelle in IDAS werden durch die sie bildenden Datenklassen und deren Beziehungen untereinander definiert. Eine typische Datenklasse des Beispiels wäre der „Teilekatalog“ der Tabelle 1.

Eine Datenklasse wird ihrerseits durch die diese Datenklasse bestimmenden Deskriptoren und deren Beziehungen untereinander gebildet. Deskriptoren des Beispiels sind „Bezeichnung“, „Komponente“, „Einkaufspreis“, „DM“, „Woche“, „Tag“ etc. (siehe Tabelle 1). Die Deskriptoren bestimmen somit die Struktur der Datenklasse. Gleiches gilt für

die Datenklassen, deren Zusammenfassung zu einem Datenmodell dessen Struktur festlegen. Allgemeiner ausgedrückt handelt es sich hier um Regeln, die die Dynamik des Datenmodells definieren.

Während des Entwurfs werden alle ermittelten Begriffe (Deskriptoren) gesammelt. Gleiche Deskriptoren werden einer Deskriptorenklasse zugeordnet (siehe Tabelle 6).

Die Bezeichnung der Deskriptorklasse selbst ist für sich betrachtet wiederum ein Deskriptor. Gleiches gilt für die Bezeichnung einer Datenklasse.

Die elementare Form der Beziehung von Deskriptoren einer Datenklasse untereinander ist die Eigenschaft, daß der Dateninhalt bestimmter Deskriptoren eine Ausprägung der Datenklasse zugehörigen Information (in der DV als Datensatz bezeichnet) eindeutig identifiziert (siehe Tabelle 7).

Es ist durchaus sinnvoll, innerhalb einer Datenklasse den Deskriptor einer anderen Datenklasse aufzunehmen, um zu einer verständlicheren Darstellung zu kommen, z.B. „Zahl“ und „Zeit“ als „Zeitdauer“ oder „Zahl“ und „Einheit“ als „Betrag in DM“.

Nachdem der Geschäftsführer und der IDAS-Experte alle Deskriptoren erfaßt, diese Deskriptorenklassen zugeordnet und danach die

Tabelle 5: Organisation und Prozesse

Organisation	Prozeß							
	Beschaffung		Produktion		Absatz		Verwaltung	
	Pla-nung	Ein-kauf	Spezi-fikation	Ter-mine	Pla-nung	Ver-kauf	Finan-zen	Perso-nal
Geschäftsführung	x	xx	(x)	(x)	x	xx	xx	xx
Mitarbeiter der/des								
- Finanzen	(x)	(x)	-	-	-	x	xx	x
- Vertriebs	x	x	(x)	-	xx	xx	(x)	x
- Entwicklung	(x)	-	x	x	(x)	-	(x)	(x)
- Produktion	-	x	xx	xx	x	x	-	(x)
- Lagers	x	-	x	-	xx	x	-	(x)
- Controllings	x	(x)	-	-	x	-	xx	x
- Einkaufs	xx	xx	-	x	(x)	-	x	-
- Verkaufs	x	-	-	x	xx	xx	x	-
- Personalabteilung	-	-	-	-	-	-	x	xx

xx = hauptverantwortlich/entscheidet mit (alle Informationen) (x) = beteiligt (wenige Informationen)
 x = stark beteiligt (wichtige Informationen) - = nicht beteiligt (keine Informationen)

Tabelle 6: Deskriptorenklassen (auszugsweise)

Zeit	Bezeichnung	Einheit	Produkt
Tag	Komponente	DM	Finanzbuchhaltung
Woche	Produkt	St. (Stück)	Netzwerk-Server-386dx
Mannjahre	Wartung/Pflege

Tabelle 7: Datenklasse „Teilekatalog“

Komponente ^a	Zahl	Zahl	Einheit	Zahl	Zeit
Monitor farbig	1	650	DM	2	Tage
Betriebssystem PC	1	150	DM	2	Tage
Disketten 5 1/4	100	100	DM	1	Woche
...

^a Die Bezeichnung der Komponente ist jeweils eindeutig

Tabelle 8: Datenklassen/Prozeß-Gruppen (auszugsweise)

Datenklasse	Prozeß					
	Unter-nehmens-planung	Produkt-spezifi-kation	Ein-kauf	Ferti-gung	Ver-kauf	Budget-rechnung
Plan	e	-	-	-	-	b
Finanzmittel	b	-	-	-	-	b
...						
Produkte	-	b	-	-	b	-
Teilekataloge	-	e	-	b	-	-
Lieferanten	-	b	e	-	-	-
Kunden	-	-	-	-	b	-

„e“ = erzeugt die Daten „b“ = benutzt die Daten

Datenklassen definiert hat, ist die Datenmodellierung abgeschlossen. Diese Vorgehensweise wird als „Bottom-Up-Strategie“ bezeichnet (= Umkehrung der „Top-Down-Strategie“, d.h. „von-unten-nach-oben“). In einem nächsten Schritt ist die Informationsarchitektur der Organisation „EDV-Systemhaus“ zu definieren. Hierzu werden die Datenklassen den Unternehmensprozessen gegenübergestellt. Es ergibt sich dabei der Sachverhalt, daß bestimmte Unternehmensprozesse Informationen (Daten) erzeugen und/oder nur benutzen (siehe Tabelle 8).

Aufgrund der festgelegten Priorität für das Anwendungsprogramm „Produktionsprogramm“ werden dessen Daten (d.h. die benötigten Datenklassen) festgelegt, um eine Datenbank innerhalb von IDAS zu erstellen. Für jede Datenklasse wird dabei eine Datenbank-tabelle erzeugt. Als einfachste Lösung für das Anwendungsprogramm wird ein Programm erstellt, das direkt mit Hilfe des definierten Datenmodells auf die Datenbank zugreift und dem Benutzer somit geordnet die Dateninhalte für den zu unterstützenden Prozeß zur Verfügung stellt.

Zusammenfassung

Anhand des einfachen Beispiels wurde aufgezeigt, daß die traditionelle Vorgehensweise, auch DV-gestützt, unstrukturiert und mit mehrfacher Datenerfassung und -bereitstellung erfolgt. Die Darstellung innerhalb von IDAS scheint nur eine strukturiertere Form für die Problemlösung zu sein. Tatsächlich können einmal erfaßte Deskriptoren und Datenklassen in anderen Datenmodellen (und damit Datenbanken) erneut verwendet werden. Eine Redundanz (Doppelspeicherung gleicher Daten) wird somit vermieden, die Entwicklung neuer Anwendungsprogramme wird erheblich beschleunigt und eine einheitliche Verwendung von Begriffen (Deskriptoren) wird automatisch gewährleistet.

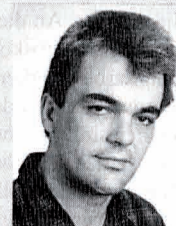
Zu den Autoren:



Prof. Dr. Dr. h.c. Miklós Géza Zilahi-Szabó, Professor für Betriebsinformatik an der Universität Gießen, seit 25 Jahren auf dem Gebiet der Informatik tätig. Hauptsächliche Beschäftigungsgebiete sind Datenmodellierungen, Datenbankentwürfe, Informationssysteme, Organisationsanalysen und Entwicklung von Anwendungsprogrammen.



Dr. Andreas van de Valk, Studium in Gießen, 1990 Promotion über Organisationsanalyse, Abteilungsleiter für Software-Entwicklung bei Systemhaus Curadata Dannenberg, der Tochtergesellschaft von debis (Daimler Benz Inter Services).



Gerd Koslowski, Studium in Göttingen, promoviert zur Zeit über diese Thematik, beschäftigt als Systemanalytiker bei HEC Hanseatische Software-Entwicklungs- und Consulting GmbH in Bremen.