

# Datenmodellierung und aufgabengerechte Dialoge: ein Synchronisationsproblem

Thomas Greutmann  
Bertelsmann Zentrale Informationsverarbeitung  
Rheda-Wiedenbrück

## Zusammenfassung

Es wird ein Ansatz vorgestellt, welcher die Bereiche Datenmodellierung und aufgaben- und benutzergerechte Gestaltung von Dialogen zusammenführt. An einem Beispiel wird eine Vorgehensweise entwickelt, welche die Gestaltung von aufgabengerechten, flexiblen Dialogen auf der Basis eines E/R-Datenmodells erlaubt. Dieser Ansatz hat sich in der Praxis als didaktischer Rahmen bewährt, um eine Mindestqualität für die Benutzungsoberfläche von Anwendungssystemen sicherzustellen und Software-Entwicklern Konzepte der Software-Ergonomie nahezubringen.

## 1 Einleitung

Methoden der Datenmodellierung kommt heutzutage beim Entwurf und beim Design von Anwendungssystemen eine entscheidende Bedeutung zu. Das Datenmodell gilt als ein wesentlicher und stabiler Kern des Anwendungssystems, dessen Güte auf alle übrigen Teile des Systems ausstrahlen kann. Dabei hat die Entity-Relationship-Methode (Chen [1]) bzw. verschiedene Varianten davon starke Verbreitung gefunden und ist ein De-Facto-Standard geworden. Für den Entwurf anderer Komponenten als der Datenbank eines Anwendungssystems werden jedoch meist andere, von E/R unabhängige Methoden verwendet. Die Konsistenz und Vollständigkeit eines Designs methodenübergreifend sicherzustellen ist eine nichttriviale, fehlerträchtige und aufwendige Aufgabe für Anwendungsentwickler.

## 2 Fragestellung: die Synchronisationsproblematik

Das Synchronisationsproblem stellt sich nicht zuletzt auch für die Benutzungsoberfläche des Anwendungssystems. Für den Entwurf von benutzer- und aufgabengerechten Dialogen gibt es eine Reihe von möglichen Methoden, u.a. Zustandsübergangsdigrammen, ATN's (Kieras & Polson [7]), RFA-Netze (Oberquelle [8]) oder Aufgabennetze (Keil-Slawik [6]), welche nicht mit Datenmodellen abgeglichen sind. Selbst ein der Software-Ergonomie wohlgesinnter Anwendungsentwickler ist also

ständig mit der Frage konfrontiert, wie er gleichzeitig und konsistent ein gutes Datenmodell und gute Dialoge entwerfen und beschreiben soll.

### 3 Die Aufgabe als Bindeglied

In der Arbeitspsychologie wird die Wichtigkeit der aufgaben- und benutzergerechten Gestaltung von Dialogsystemen betont (z.B. Hacker [5] oder Ulich [11]). Diese Forderung ist in Form des Kriteriums "Aufgabenangemessenheit" sogar in der DIN-Norm 66234 Teil 8 [2] aufgenommen. Die Operationalisierung dieses Kriteriums "Aufgabenangemessenheit" in einer für Software-Entwickler anwendbaren Form hat sich bisher als schwierig dargestellt (Rödiger & Piepenburg [10]).

Eine für die Zwecke der Software-Entwicklung sinnvolle und meines Erachtens auch für Software-Entwickler verständliche Operationalisierung lehnt sich an Hackers ([4], S.69) Definition von Aufgabe an. Danach wird eine Aufgabe durch fünf Merkmale beschrieben (vgl. Greutmann [3], S.47):

1. Die *Benutzer*, welche die Aufgabe bearbeiten.
2. Die *Abläufe* und *Vorgehensweisen*, welche bei der Aufgabenbearbeitung befolgt werden müssen.
3. Die *Objekte*, welche dabei benutzt oder verändert werden.
4. Die *Methoden* und *Werkzeuge*, welche zur Unterstützung benötigt werden.
5. Die äusseren *Rahmenbedingungen*, unter denen die Aufgabe durchgeführt wird.

Grundlage für die software-ergonomische Gestaltung eines Anwendungssystems stellt eine vollständige Aufgabenbeschreibung dar, welche für alle mit dem System zu bewältigenden Aufgaben alle fünf Komponenten beinhaltet.

Die Verbindung zwischen E/R-Datenmodellierung und Aufgabenbeschreibungen erfolgt über eine gemeinsame Komponente: die *Objekte*, welche in E/R als Entitäten auftauchen. Die Grundstruktur des Dialogs wird entsprechend der Struktur des E/R-Datenmodells ausgelegt - dies setzt ein mit den Anwendern zusammen entwickeltes, aufgabenorientiertes Datenmodell voraus: Ein gutes Datenmodell ist immer auch aufgabenangemessen.

### 4 Die Rolle von Benutzern, Kunden und Anwendern

Für die gesamte Vorgehensweise gilt, daß sie nicht alleine von Software-Entwicklern durchgeführt werden darf. Sowohl für die Datenmodellierung wie auch für die Dialoggestaltung müssen als Mindestanforderung die entsprechenden Wissens- und

Entscheidungsträger einbezogen werden. Je nach Fragestellung und Ebene handelt es sich dabei um verschiedene Personengruppen (vgl. Heilmann 1981, zit. in Peschke [9]). Im konkreten Fall bedeutet dies:

- Für die Datenmodellierung müssen vor allem die Wissensträger einbezogen werden, welche das Gesamtgeschäft überblicken, z.B. Führungskräfte und Fachspezialisten des Anwenders, u.U. auch zukünftige System-Dienstleistungsempfänger.
- Für aufgabengerechte Dialoge sind in erster Linie die zukünftigen Endbenutzer einzubeziehen.

## 5 Die Vorgehensweise

### 5.1 Voraussetzungen

Es wird vorausgesetzt, daß vorgängig unter Einbeziehung der jeweiligen Wissensträger ein vollständiges logisches E/R-Datenmodell erstellt wurde. Dies bedeutet - ohne auf Einzelheiten der Datenmodellierung einzugehen - u.a.:

1. Das Datenmodell ist vollständig normalisiert und redundanzfrei.
2. Alle komplexen Beziehungen (N zu N, Kann zu Kann etc.) sind durch Hinzufügen zusätzlicher Entitäten aufgelöst, so daß nur noch Beziehungen der Form 1-zu-irgendetwas übrigbleiben.

Abb. 1. zeigt ein solches Datenmodell für ein Seminarbüro-Reservierungssystem.

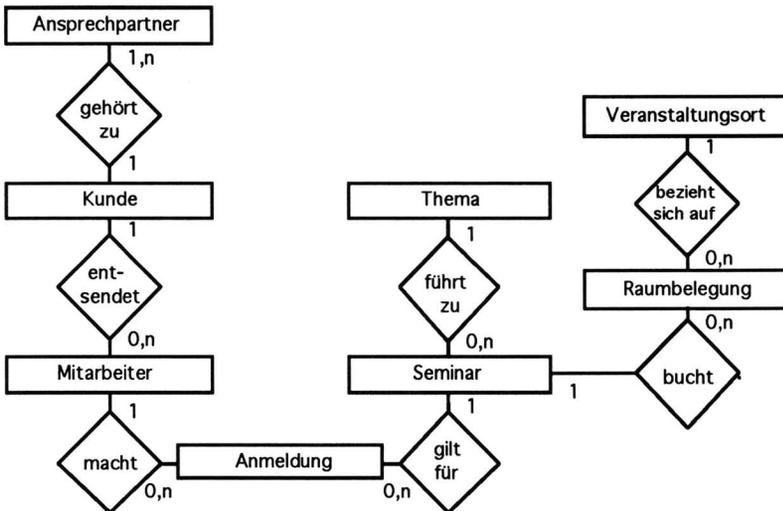


Abb. 1: Logisches Datenmodell für ein Seminarbüro-Reservierungssystem (vereinfacht).

Weiter wird vorausgesetzt, daß eine vollständige Aufgabenbeschreibung mit allen Komponenten für alle Aufgaben (siehe oben) unter Einbezug mindestens der Endbenutzer erstellt wurde.

## 5.2 Erstellung eines flexiblen Navigationsdialogs

Dies ist der zentrale Schritt, durch welchen die Aufgabenbeschreibungen und das Datenmodell synchronisiert werden. Ausgehend vom logischen Datenmodell und den Aufgabenbeschreibungen wird ein flexibler Navigationsdialog durch den gesamten Datenraum erstellt. Dazu gehören folgende Teilschritte:

1. Bestimmung der Informationsbildschirme gemäß Datenmodell
2. Festlegung der Bewegungspfade zwischen den Informationsbildschirmen
3. Aufgabengerechte Festlegung des Inhalts der einzelnen Informationsbildschirme
4. Festlegung der Einstiegspunkte für flexible Navigation

### 5.2.1 Bestimmung der Informationsbildschirme gemäß Datenmodell

Ein Informationsbildschirm ist - abhängig von der verwendeten Technologie - eine Bildschirmseite oder ein Dialogfenster. Bei umfangreichen Datenmengen können auch mehrere Seiten bzw. Fenster benötigt werden. Die Regel für die Bestimmung der Informationsbildschirme ist einfach:

Regel 1: Grundsätzlich wird für jede Entität im logischen Datenmodell ein Informationsbildschirm angelegt.

Allenfalls kann für einzelne Entitäten, welche geringe Datenmengen enthalten und welche durch Normalisierungen entstanden sind, auf einen eigenen Informationsbildschirm verzichtet werden. Diese Entitäten sind meistens hierarchisch von einer anderen Entität abhängig und können dort integriert werden. Dies wird aber erst im Teilschritt 2.3 offensichtlich. Im Beispiel wird für die Entität "Ansprechpartner" kein eigener Bildschirm benötigt (vgl. Abb. 1), weil alle Daten auf dem Informationsbildschirm für "Kunde" dargestellt werden können.

### 5.2.2 Festlegung der Bewegungspfade zwischen den Informationsbildschirmen

Die Grundregel für Bewegungspfade zwischen Informationsbildschirmen ist ebenfalls einfach:

**Regel 2:** Von jedem Informationsbildschirm aus müssen alle Informationsbildschirme angewählt werden können, welche im logischen Datenmodell miteinander in Beziehung stehen.

Dadurch wird es dem Benutzer ermöglicht, von einem Objekt aus alle anderen Objekte zu erreichen, welche dazu in Beziehung stehen. Im Beispiel müssen von einem Objekt "Seminar" aus das zugehörige Thema, alle Raumbelagungen sowie alle dazu erfolgten Anmeldungen gewählt werden können.

### 5.2.3 Aufgabengerechte Festlegung des Inhalts der einzelnen Informationsbildschirme

In diesem Teilschritt werden logisches Datenmodell und Aufgabenbeschreibung miteinander verknüpft. Die Regeln sind entsprechend vielfältiger.

**Regel 3:** Jeder Informationsbildschirm stellt mindestens die Attribute der jeweiligen Entität dar.

Diese Regel ist eigentlich trivial und läßt sich an einem Beispielschirm (Abb. 2) verdeutlichen. Dort werden auf einem Seminarbildschirm die Informationen zur Entität Seminar (z.B. Datum) in der Mitte dargestellt.

**Regel 4:** Darüberhinaus werden entlang der Beziehungen die Informationen aus benachbarten Entitäten dargestellt, welche zur aufgaben- und benutzergerechten Darstellung notwendig sind.

In Datenbank-Terminologie bedeutet dies: jeder Informationsbildschirm stellt nicht eine Entität, sondern eine View, d.h. einen zusammenhängenden Ausschnitt aus dem Datenmodell dar. Damit werden die aus datenbanktechnischen Gründen erwünschten Normalisierungen für die Benutzer unsichtbar gemacht. Dies ist im Beispiel in Abb. 2 ersichtlich, wo zum Seminar auch Informationen der Entitäten Thema (oben), Anmeldung und Raumbelegung (unten) dargestellt werden, die mit dem angezeigten Seminar-Objekt in Beziehung stehen.

Es lassen sich auch Ergänzungsregeln zu Regel 4 formulieren, welche klarer den Zusammenhang zwischen Aufgabenbeschreibung und Datenmodell zeigen:

**Regel 4.1:** Von allen benachbarten Entitäten im Datenmodell müssen so viele Attribute dargestellt werden, daß die dazu in Beziehung stehenden Objekte für den Benutzer verständlich sind.

So genügt es im Beispiel für den Informationsbildschirm "Seminar", zum Thema den Titel aufzuführen (Abb. 2, N-zu-1-Beziehung zu Thema). Dies ist für die Benutzer verständlich. Außerdem muß die Liste aller Anmeldungen dargestellt werden (1-zu-N-Beziehung zu Anmeldung).

Abb. 2: Beispiel für einen Informationsbildschirm zur Entität "Seminar". Dieser stellt auch Informationen von den Entitäten Thema, Anmeldung und Mitarbeiter dar.

Regel 4.2: Stehen die benachbarten Entitäten im Datenmodell ihrerseits in hierarchischen (N-zu-1 bzw. auch 1-zu-1) Beziehungen zu weiteren Entitäten, so müssen entlang solcher Hierarchien soweit Attribute von weiteren Entitäten dargestellt werden, bis die Darstellung benutzer- und aufgabengerecht ist. Diese Regel kann wiederholt angewendet werden.

Dies läßt sich am Beispiel in Abb. 2 verdeutlichen. Dort werden die Namen der angemeldeten Mitarbeiter zu einem Seminar dargestellt. Diese Namen sind jedoch nicht in der unmittelbar benachbarten Entität Anmeldung enthalten (vgl. Datenmodell in Abb. 1), sondern in der Entität Mitarbeiter, welche in einer N-zu-1-Beziehung zu Anmeldung steht.

Es könnte durchaus sein, daß dies dem Benutzer noch nicht genügt. Die Mitarbeiter stehen ihrerseits in N-zu-1-Beziehung zu "Kunde", so daß es notwendig sein kann, auf dem Informationsbildschirm Seminar (Abb. 2) zusätzlich zum Mitarbeiter Kundeninformationen darzustellen (z.B. den Firmennamen). Wie oft Regel 4.2 ange-

wendet werden muß, läßt sich nicht aus dem Datenmodell ableiten, sondern ist mit dem Endbenutzer zu klären: nur er kann entscheiden, was aufgabenangemessen ist.

**Regel 5:** Die Art der Darstellung von Informationen auf Informationsschirmen muß aufgaben- und benutzergerecht sein.

Auch diese Regel läßt sich am besten anhand des Beispiels erklären. Für den Informationsbildschirm "Veranstaltungsort" lassen sich die dort abgehaltenen Seminare in einer Liste darstellen. Besser für die Benutzer, welche Seminare disponieren müssen, wäre aber ein Belegungsplan in Kalenderdarstellung. Auch dies läßt sich nur anhand der Aufgabenbeschreibungen zusammen mit dem Benutzer eindeutig klären und geht nicht aus dem Datenmodell hervor.

#### 5.2.4 Festlegung der Einstiegspunkte für flexible Navigation

Bisher wurde für den Navigationsdialog vorausgesetzt, daß stets ein Objekt ausgewählt ist. Um die Navigation zu beginnen, müssen dem Benutzer aber zusätzlich flexible Einstiegsmöglichkeiten in Form von Auswahllisten und Suchmasken (Eingabe von beliebigen Suchkriterien) geboten werden. Mit Hilfe dieser Einstiegspunkte kann ein Objekt bestimmt werden, von welchem aus der Navigationsdialog begonnen werden kann. Solche Einstiegsmasken sind nur für bestimmte Entitäten sinnvoll:

**Regel 6:** Erste Kandidaten für Einstiegsmasken sind die Entitäten, welche an der Spitze von Hierarchien stehen, d.h. von denen aus nur 1-zu-x-Beziehungen ausgehen.

Im Datenmodell von Abb. 1 gilt dies für die Entitäten "Kunde" und "Thema". Über diese Einstiegspunkte kann der gesamte Datenraum traversiert werden. Diese Regel genügt aber im allgemeinen nicht. Es lassen sich noch weitere Kandidaten benennen:

**Regel 6.1:** Weitere Kandidaten für Einstiegspunkte sind Entitäten, welche einen eigenen, d.h. nicht vererbten Schlüssel haben.

Im Datenmodell von Abb. 1 würde die Entität "Mitarbeiter" darunterfallen, da der Schlüssel im Beispiel eine eigene Mitarbeiter-Id ist, welche nicht von Kunde vererbt ist. Für "Seminar" würde dagegen keine Einstiegsmaske vorgesehen, weil der Schlüssel teilweise von "Thema" vererbt ist (Themen-Kürzel oben in Abb. 2 ist Teil des Seminar-Schlüssels).

Um die Navigation für die Benutzer so flexibel wie möglich zu gestalten, müssen die Einstiegspunkte jederzeit angewählt werden können - egal, welcher Informationsbildschirm gerade aktiv ist. Dadurch wird die Navigation beliebig unterbrechbar. Dies kann z.B. durch Einbettung der Einstiegspunkte als Menüpunkte in ein ständig verfügbares Pull-Down-Menü erfolgen.

### 5.3 Ergänzung der Navigationsdialogs-Masken

Mit diesem Navigationsdialog kann flexibel der ganze Datenraum inspiziert werden, es sind aber noch keine Änderungen des Datenbestands möglich. Diese Möglichkeiten werden nun im nächsten Entwurfsschritt ergänzt:

1. *Änderungsmöglichkeiten auf Entitätenebene:* Dazu gehören: Ändern von Attributen der Entität, Erzeugen von neuen Objekten und Löschen von Objekten. Im Beispiel der Seminarmaske (Abb. 2) sind diese als Buttons realisiert (Mitte rechts), welche spezielle Änderungsmasken bzw. Bestätigungsmeldungen aufrufen. Dadurch können neue Seminare angelegt, Seminare gelöscht oder z.B. das Datum geändert werden.
2. *Änderungsmöglichkeiten auf Beziehungsebene:* Die Beziehungen zu anderen Objekten müssen gemäß Datenmodell verändert werden können. Beziehungen können entfernt bzw. neu hinzugefügt werden. In der Seminarmaske (Abb. 2) sind diese Änderungsmöglichkeiten in die Änderungsmaske für die Entität (Button "Ändern") integriert. Dadurch können neue Mitarbeiter angemeldet werden bzw. die Veranstaltungsorte geändert werden.
3. *Weitere aufgabenspezifische Änderungsmöglichkeiten.* Mit Hilfe der oben beschriebenen Ergänzungen des Navigationsdialogs erhält man ein Anwendungssystem, dessen Funktionalität aus technischer Sicht vollständig ist, d.h. alle Informationen können abgefragt und die Datenbestände beliebig verändert werden. Die wesentlichen Arbeitsabläufe jedoch, welche in den Aufgabenbeschreibungen dokumentiert sind, werden durch diese Änderungsmöglichkeiten, die sich am Datenmodell orientieren, nicht adäquat abgedeckt. Deshalb werden die Informationsbildschirme des Navigationsdialogs durch weitere Funktionalität ergänzt, welche den Aufgaben der Benutzer entsprechen.

Dies wird am Beispiel in Abb. 2 deutlich. Rechts unten sind einige Buttons hinzugefügt, hinter welchen sich aufgabenspezifische Funktionalität verbirgt. Hinter dem Button "Teilnahme vermerken" verbirgt sich eine weitere Maske, auf welcher die Teilnahmekontrolle ("Abhaken" der erschienenen Mitarbeiter) durchgeführt werden kann. Ähnliche Masken sind zur Überprüfung der Rechnungsbeträge (Button "Rechnungen prüfen"), zur Anmeldung von Teilnehmern (Button "Neuanmeldung") und zur Stornierung des Seminars (Button "Seminar stornieren") vorgesehen.

## 5.4 Weitere aufgabenspezifische Ergänzungen

Der bis hier entstandene Dialog ist eigentlich "objektorientiert": im Zentrum stehen die Datenobjekte, zwischen denen navigiert werden kann. An die Objekte gekoppelt sind die aufgabengerechten Funktionen, welche auf ein Objekt angewandt werden können.

Dies ist zwar bereits ein sehr komfortables Gerüst für die Benutzungsschnittstelle eines Anwendungssystems, genügt aber u.U. immer noch nicht völlig für einen vollständig aufgabengerechten Dialog. Es gibt Aufgaben bzw. Aufgabenbestandteile, welche sich nicht immer sinnvoll an ein Datenobjekt koppeln lassen, sondern welche eher ereignisorientiert sind. Im Beispiel wäre dies eine Anmeldefunktion, welche jederzeit und rasch (z.B. bei einem Telefonanruf) ohne Navigation aktivierbar sein muß. Solche Sachverhalte müssen entsprechend in den Aufgabenbeschreibungen dokumentiert sein.

Der Dialog muß dementsprechend erweitert werden. Es handelt sich dabei ebenfalls um Einstiegspunkte, die aber im Gegensatz zu den Einstiegspunkten für den Navigationsdialog nicht objektbezogen, sondern ereignisbezogen sind. Analog werden solche Einstiegspunkte z.B. als Menüpunkte in einem permanenten Pull-Down-Menü realisiert, hinter welchen sich aufgabenspezifische Masken verbergen, die mit dem Datenmodell nicht ohne weiteres in Beziehung zu setzen sind.

# 6 Diskussion

## 6.1 Eigenschaften der Dialoge

Dialoge für Anwendungssysteme, welche nach der hier vorgestellten Vorgehensweise konzipiert werden, haben die folgenden Eigenschaften:

- Sie bauen auf einem *flexiblen Navigationsdialog* auf, mit dessen Hilfe der gesamte Datenraum mit all seinen Objekten und Beziehungen in einer für den Benutzer bedeutungsvollen Weise traversiert werden kann.
- Die Navigationsmöglichkeiten werden durch spezifische Änderungsfunktionen ergänzt, welche den Objekten zugeordnet sind. Dies resultiert in einem *objektorientierten, aufgabengerechten Dialog*.
- Wo nötig, wird der Dialog ergänzt durch *ereignisorientierte Komponenten*.

Solche Dialoge sind besonders geeignet für Informationssysteme, welche Aufgaben mit sowohl *operativen* als auch *dispositiven* Komponenten unterstützen sollen. Die flexible Navigation in Verbindung mit einem objektorientierten Dialog unterstützt die dispositiven Komponenten, die ereignisorientierten Ergänzungen eher die operativen Komponenten. Die

Dialoge sind flexibel und zwingen Benutzer nicht in das Korsett eines standardisierten Ablaufs - wichtig vor allem für regelmäßige Benutzer.

## 6.2 Die Vorgehensweise als didaktisches Konzept

In dieser Vorgehensweise wird ein technisches Hilfsmittel, die Datenmodellierung, mit der Gestaltung von aufgaben- und benutzergerechten Dialogen verbunden. Dabei wird eine Definition von "Aufgabe" verwendet, welche sich in der Diskussion mit Entwicklern und in Schulungen als verständlich und hilfreich erwiesen hat. Durch den Bezug auf das den meisten Entwicklern gängige methodische Hilfsmittel "Datenmodell" werden Entwickler in ihrer Sprache angesprochen und damit durch die Hintertür für die Probleme der aufgaben- und benutzergerechten Gestaltung von Anwendungssystemen sensibilisiert, ohne daß sie dabei durch den Begriff "Software-Ergonomie" erschreckt werden müssten.

## 6.3 Vorteile der Vorgehensweise

Die Befolgung dieser Vorgehensweise bringt in mehrerer Hinsicht Vorteile:

- Zwei scheinbar unabhängige Gebiete, Datenmodellierung und Gestaltung von Dialogen können konsistent miteinander verknüpft werden.
- Ein komfortabler, flexibler, benutzer- und aufgabengerechter Dialog für den gesamten Datenraum wird erstellt. Dies sichert eine Art Mindestqualität des Dialogs, der Dialog enthält Freiheitsgrade für dispositive Möglichkeiten, eine Eigenschaft, die auch aus Gründen der Arbeitsgestaltung interessant ist: solche Dialoge sichern dem Benutzer dispositive Möglichkeiten, auch wenn dies in den Aufgabenbeschreibungen nicht explizit vorgesehen wurde.
- Das E/R-Datenmodell, welches in seinen Grundzügen oft in Projekt-Vorphasen erstellt wird, wird durch aufgabenbezogenes Dialogdesign überprüft (Qualitätssicherung). Dies wird besonders unterstützt durch den Einsatz von simulationsfähigen Prototyping-Werkzeugen, mit deren Hilfe die Dialoge für Benutzer visualisiert werden können.

## 6.4 Entscheidende Erfolgsfaktoren

Die hier vorgestellte Methode stellt nicht einen sklavisch zu befolgenden Algorithmus für die Gestaltung von Dialogsystemen dar, sondern soll in erster Linie eine Reihe von Anregungen vermitteln. Die Güte der mit Hilfe dieser Vorgehensweise erstellten Dialoge hängt ganz wesentlich von den Voraussetzungen, d.h. von der Güte des Datenmodells und der Aufgabenbeschreibungen ab. Diese können nur in Zusammenarbeit mit den jeweiligen Wissens- und Entscheidungsträgern erarbeitet werden.

Die Güte dieser Zusammenarbeit bestimmt die Güte der Ergebnisse wesentlich. Für den Software-Ergonom in der Praxis bedeutet dies, daß er nicht nur über Dialoge Bescheid wissen muß, sondern u.a. viel von Datenmodellierung verstehen sollte, um die entsprechende Zusammenarbeit mit den Wissens- und Entscheidungsträgern kompetent durchführen zu können. Ein gutes Datenmodell ist stets auch aufgabengerecht und eine wesentliche Grundlage für Dialogdesign.

## 7 Literatur

- [1] Chen, P.P. (1976). The Entity-Relationship Model - Toward a Unified View of Data. ACM Transactions on Database Systems 1(1), S.9-36.
- [2] DIN 662324 Teil 8 (1988). Bildschirmarbeitsplätze - Grundsätze ergonomischer Dialogsysteme. Beuth, Berlin.
- [3] Greutmann, T. (1992). HIDE and IDEA: Tools for User-Oriented Application Development. Informatik-Dissertationen ETH Zürich, Nr. 35. Verlag der Fachvereine, Zürich.
- [4] Hacker, W. (1986). Arbeitspsychologie - Psychische Regulation von Arbeitstätigkeiten. VEB Deutscher Verlag der Wissenschaften, Berlin.
- [5] Hacker, W. (1987). Software-Ergonomie: Gestalten rechnergestützter geistiger Arbeit?!, in: W. Schönplflug & M. Wittstock (Hrsg.). Software-Ergonomie '87. Nützen Informationssysteme dem Benutzer? Teubner, Stuttgart S.31-54.
- [6] Keil-Slawik, R. (1989). Systemgestaltung mit Aufgabennetzen, in: S. Maaß & H. Oberquelle (Hrsg.). Software-Ergonomie '89. Aufgabenorientierte Systemgestaltung und Funktionalität. Teubner, Stuttgart, S.123-133.
- [7] Kieras, D. & Polson, P.G. (1983). A Generalized Transition Network Representation for Interactive Systems. Proceedings of the CHI 1983.
- [8] Oberquelle, H. (1987). Sprachkonzepte für benutzergerechte Systeme. Springer, Berlin.
- [9] Peschke, H. (1988). Partizipative Entwicklung und Einführung von Informationssystemen, in: H. Balzert, H.U.Hoppe, R. Oppermann, H. Peschke, G. Rohr & N.A. Streitz (Hrsg.). Einführung in die Software-Ergonomie. De Gruyter, Berlin, S.299-322.
- [10] Rödiger, K.-H., Piepenburg, U. (1989). Prüfung von Software auf die Grundsätze ergonomischer Dialoggestaltung, in: S. Maaß & H. Oberquelle (Hrsg.). Software-Ergonomie '89. Aufgabenorientierte Systemgestaltung und Funktionalität. Teubner, Stuttgart, S.163-173.
- [11] Ulich, E. (1989). Arbeitspsychologische Konzepte der Aufgabengestaltung, in: S. Maaß & H. Oberquelle (Hrsg.). Software-Ergonomie '89. Aufgabenorientierte Systemgestaltung und Funktionalität. Teubner, Stuttgart, S.51-65.

Dr. Thomas Greutmann  
Bertelsmann Zentrale Informationsverarbeitung  
Steinweg / Schloß Rheda  
W-4840 Rheda-Wiedenbrück  
Tel. 05242-415 855

