

Zeitschrift:	Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses
Herausgeber:	Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen
Band:	81 (1990)
Heft:	13
Artikel:	Technische Informationssysteme bei den Schweizer Bundesbahnen : Systemintegration der Fahrplan-Datenbank
Autor:	Hügli, Ulrich
DOI:	https://doi.org/10.5169/seals-903137

Nutzungsbedingungen

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

Conditions d'utilisation

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

Terms of use

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

Download PDF: 14.01.2026

ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>

Technische Informationssysteme bei den Schweizerischen Bundesbahnen

Systemintegration der Fahrplan-Datenbank

Ulrich Hügli

Die Notwendigkeit, Datenbankanwendungen nicht isoliert, sondern systemübergreifend zu planen, ist heute kaum mehr umstritten. Die Frage, ob und wie weit auch die Daten von technisch orientierten Systemen in diese Integrationsbemühungen einbezogen werden sollen oder können, wird im vorliegenden Artikel am Beispiel der SBB-Fahrplan-Datenbank diskutiert.

La nécessité d'une intégration des bases de données qui ne se confine pas aux systèmes d'informations isolés n'est plus guère contestée. Cet article traite de la question, à quel point les systèmes avec un champ d'application purement technique peuvent faire partie de cette intégration.

Der Übergang von der traditionellen, ablauforientierten Datenverarbeitung zu Applikationen, welche auf Daten zugreifen, die in Datenbanken organisiert sind, hat neue Vorgehensweisen für den Entwurf von Datenstrukturen notwendig gemacht. Wurden ehemals die Daten nur in Hinsicht auf eine einzige Verarbeitung oder ein einzelnes Programm strukturiert, so werden im Zusammenhang mit dem Datenbank-Entwurf Methoden angewandt, welche erlauben, gleichzeitig die Erfordernisse mehrerer Verarbeitungen (Applikationen) zu berücksichtigen. Im Zuge der Loslösung der Datenstrukturen von den Einzelprogrammen konnte der folgende, durchaus gewünschte Nebeneffekt erzeugt werden: Es können nun nicht nur Informationen über die Struktur von Datenbeständen programmunabhängig formuliert und festgehalten, sondern auch eine grosse Klasse von Konsistenzbedingungen¹ von den Verarbeitungsbeschreibungen getrennt werden. Diese Konsistenzbedingungen bilden fortan einen wichtigen Bestandteil der Datenbank-Beschreibung, und ihre Einhaltung wird vom Datenbank-Management-System (DBMS) selbsttätig überwacht. Die Fortsetzung der Zentralisierung von Informationen über die logische Korrektheit und das Verhalten von Datenbeständen war in der Folge in zwei Richtungen zwangsläufig: Erstens musste früher oder später die Frage gestellt werden, weshalb Datenbestände zwar programmübergreifend, nicht aber in einem systemübergreifenden Rahmen gesehen und strukturiert werden.

Zweitens musste irgendwann zur Überwindung des Dualismus Daten-

beschreibung und Verarbeitungsbeschreibung angesetzt werden.

Die Antwort auf die erste Frage hat zu neuen Verfahren der Systementwicklung geführt. Das von den SBB im Bereich der Datenstrukturierung gewählte Verfahren, welches in verschiedenen Systemen redundante Beschreibungen der gleichen Strukturbereiche ausschliessen soll, wird im folgenden kurz erläutert. Die zweite Folgerung hat zum Postulat der objektorientierten Datenbanken geführt. Auf diese soll hier nicht näher eingegangen werden.

Datenintegration als Richtlinie bei der Entwicklung von Informationssystemen

Die zentrale Schwierigkeit bei der bezüglich Datenplanung integrierten Entwicklung einer Vielzahl von Systemen lag bei der Wahl der *Darstellung der Daten* und ihrer *Bedeutungen*. Vorerst war also ein geeignetes Datenmodell zu finden, welches eine Kommunikation über die Systemgrenzen hinweg erlaubt. Als Verständigungsgrundlage wurde eine sogenannte Entity-Relationship-Darstellung gewählt, welche Daten und Bedeutungen oder kurz Informationen in einer transparenten Weise darzustellen vermag. Im wesentlichen bringen solche Entity-Relationship-Diagramme zum Ausdruck, welche Klassen von Informationsobjekten in einem bestimmten System von Interesse sind und welche Typen von Beziehungen zwischen den Repräsentanten dieser Klassen bestehen können. Die Bedeutungen und klaren begrifflichen Umschreibungen sowohl der *Informationsobjekt-Klassen* als auch der *Beziehungs-Typen* bilden den zentralen Beschreibungsteil.

Adresse des Autors

Ulrich Hügli, Dipl. Phys., Schweizerische Bundesbahnen, Direktion Informatik, Bollwerk 10, 3030 Bern

¹Darunter sind all diejenigen Bedingungen zu verstehen, deren Einhaltung die logische Korrektheit der Datenbestände garantiert.

Die einzelnen Systeme formulieren ihre Bedürfnisse bezüglich Datenhaltung also auf einer rein logischen Ebene, logisch in dem Sinne, als keinerlei für eine Maschine oder auch nur für ein Datenbank-Management-System verwertbare Angaben gemacht werden. Diese Darstellungen der Einzelsysteme werden zu einer sogenannten Datenarchitektur integriert (Bild 1).

lose Dokumentation der datenplanerischen Ergebnisse in einem *Data-Dictionary-System* gegeben. Die Vermeidung von Synonymen obliegt einem zentralen Dienst.

Jedes System kann gewisse Teile seiner Datenbasis als schon vorhandene voraussetzen (Deduktion), bereichert aber in der Regel die Datenarchitektur durch weitere Strukturteile (Induktion).

ship-Darstellung gewonnen werden können. Eine kurze Beschreibung der Theorie der Darstellung von Datenstrukturen durch Relationen findet sich im Kasten S. 19.

Die Sichten (Views), welche einzelne Verarbeitungen des Systems auf die Datenbasis haben, tragen zu einer Detaillierung der Relationen bei. Durch die Erweiterung des Zugriffsmodells um dynamische Aspekte entsteht ein konzeptionelles Strukturdiagramm, welches den Ausgangspunkt für die Umsetzung der Datenbasis in ein Datenbank-Management-System bildet (Bild 1). Für jede Ausprägung von Datenbank-Systemen existiert dabei ein Satz von Regeln, welche für eine systematische Umsetzung zu beachten sind. Die Entwicklung einer Datenbank-Struktur muss gegebenenfalls noch um eine Verteilungsplanung ergänzt werden.

Die Vorteile einer integrierten Datenführung

Wird ein derartiger Aufwand für die Datenintegration innerhalb der Systementwicklung betrieben, so müssen integrierte Datenbestände handfeste Vorteile für die Unternehmung aufweisen. Zum einen wird ein grosser Nutzen erzielt durch die Vermeidung oder wenigstens die Kontrolle von Redundanzen, d.h. von mehrfacher Abspeicherung gleicher Fakten. Eine redundante Datenführung, welche nicht kontrolliert werden kann, führt immer früher oder später zu widersprüchlichen Aussagen. Datenbestände, welche systemübergreifend entwickelt werden, sind also kompakter und widerspruchsfrei wartbar.

Einen weiteren Vorteil von wachsender Bedeutung gewinnt die Unternehmung bei der Bewirtschaftung integrierter Datenbestände durch die Möglichkeit, längere Informationsketten zu bilden. Diese Eigenschaft kommt durch die gegenseitige Datentransparenz zwischen den einzelnen Systemen zustande. Für die SBB ist dieser Aspekt gegenüber ihren Kunden natürlich von herausragender Bedeutung. Der zwischen den Systemen koordinierte Aufbau der Datenbestände führt zu einem Ergebnis, welches unabhängig von jeder maschinellen Datenverarbeitung unmittelbar zu einer Produktivitätssteigerung führen kann: Im Zuge der auf einem sehr abstrakten logischen Niveau geführten Diskussion der Begriffe im Zusammenhang mit Informationsobjekten

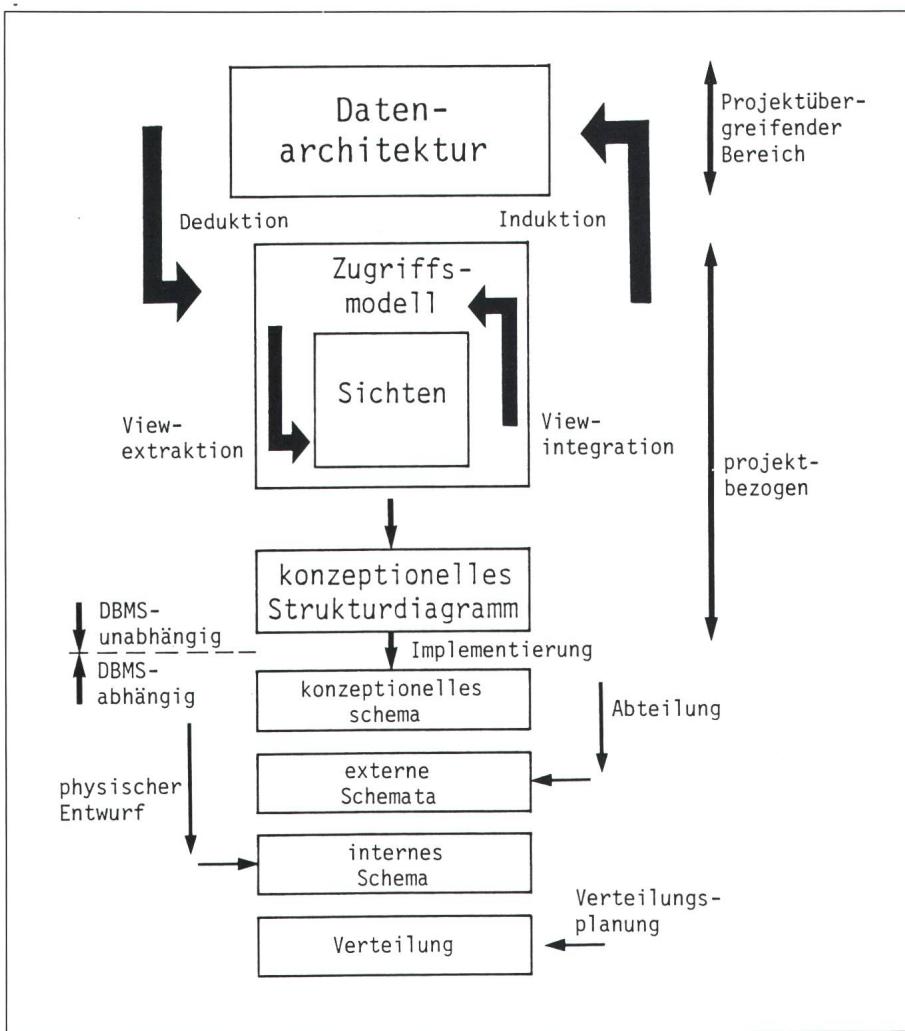


Bild 1 Integrierter Datenbank-Entwurf

Dieser Integrationsprozess ist nur möglich, wenn einerseits die bereits beschriebenen Teile der Datenarchitektur für jedes neu zu entwickelnde System in einem zentralen Medium zugänglich sind und andererseits jeder neue Eintrag auch wirklich neu ist, also insbesondere das Problem der Vergabe von Synonymen gelöst ist. Die Garantie für das Gelingen der Datenintegration ist durch die ausnahms-

Der weitere Entwurf der Datenbestände bezieht Beschreibungsebenen ein, die sukzessive relevanter für das Ziel-DBMS werden. Der für ein einzelnes System sichtbare Bereich der Datenarchitektur, das Zugriffsmodell, wird auf der Grundlage des rationalen Datenmodells beschrieben, dessen Darstellungselemente, die Relationen, aufgrund von Transformationsregeln aus den Elementen der Entity-Relation-

Klassen wird so manche fällige Begriffsdefinition endlich erarbeitet und unternehmensweit als verbindlich erklärt.

Das System Fahrplan

Am Beispiel eines in der letzten Zeit noch als Insellsolution fertiggestellten Informationssystems sollen Notwendigkeit und praktischer Nutzen der Datenintegration aufgezeigt werden: Syfa, das System Fahrplan, ist ein Direktzeit-Planungssystem zur EDV-unterstützten Konstruktion von Fahrord-

tenbank. Als Datenbank wurde DB2 von IBM gewählt, welches das relationale Datenmodell unterstützt.

Die graphischen Konstruktionselemente, temporär in der Form graphischer Vektoren abgespeichert, müssen in numerische Daten umgewandelt und der Datenbank übergeben werden. Das System Syfa – soviel geht aus dieser kurzen Systembeschreibung hervor – reiht sich in jene klassischen Ingenieurapplikationen ein, bei denen Algorithmen im Zentrum stehen, für deren Entwicklung ein beträchtliches Mass an Kreativität aufgewandt wird.

mitunter beinahe jedes noch zu entwickelnde Informationssystem auf Daten von Syfa angewiesen ist. Dies ist insbesondere der Fall für eine ganze Gruppe von Kundeninformationssystemen, die Bewirtschaftungssysteme für das Rollmaterial, die Systeme für die innerbetriebliche Verständigung, um nur einige der in Entwicklung begriffenen Systeme zu nennen. Weiterhin ist zu bemerken, dass Fahrplandaten auch eine gewichtige Rolle in den Systemen der Baudienste, so in Steuersystemen für Betriebsleitzentralen usw., spielen, welche eher den Prozesssteuerungs-Systemen zuzurechnen sind. All diese Systeme sollten nun aus einer gemeinsamen Datenarchitektur heraus entwickelt worden sein, was nun aber vorerst ein «Rückwärts-Design» von Syfa selbst erfordert. Das heisst, die relationale Darstellung der Daten aus Syfa muss auf die ursprünglichere Darstellungsform des Entity-Relationship-Diagrammes zurückgeführt werden. Im Zuge dieser Rückwärts-Herleitung müssen Begriffe und Bedeutungen nachträglich ganz klar gefasst werden. Zum Beispiel muss klar herausgearbeitet werden, welche Objekte die Klasse Betriebspunkte repräsentieren dürfen (Bild 2).

Die Integration mit dem System für betriebliche Verständigung auf der Ebene der Datenarchitektur kann etwa die folgende Erweiterung ergeben (Bild 3).

Das Vorgehen der Systementwicklung muss natürlich sicherstellen, dass die Aussagen über Zusammenhänge bis in den Bereich der physischen Be-

Das relationale Datenmodell

Das relationale Datenmodell von E.F. Codd kann durch die folgenden drei Eigenschaften umschrieben werden:

1. Die Strukturelemente sind einfach und basieren auf der Mengenlehre. Äußerlich stellen sie sich als zweidimensionale Tabellen dar, deren Bausteine aus Wertemengen, auch Domänen genannt, stammen.

Eine Kolonne einer Relation heisst Attribut. Zu einem Attribut wird eine Domäne definiert. Eine Zeile einer Relation heisst Tupel und bildet eine logische Verbindung der Informationselemente, welche durch die Attributwerte repräsentiert werden. Jede Information der Datenbasis muss explizit durch Tabellenwerte dargestellt werden. Im besonderen existieren keine Zeiger von einer Relation auf eine andere.

2. Mehrere Operatoren, welche auf die Relationen angewandt werden und deren Gesetze durch die Relationenalgebra beschrieben werden, erlauben die Suche und die Pflege der Daten. Jeder Operator wirkt auf eine oder zwei Relationen und produziert eine weitere Relation (eventuell nur aus einem Tupel bestehend).

Die Relationenalgebra umfasst klassische Operatoren der Mengenlehre (Kartesisches Produkt, Durchschnitt usw.) wie auch Operatoren, welche Untermengen von einer oder zwei Relationen (Selektion, Projektion, Theta-Produkt usw.) ergeben.

3. Eine Menge von Konsistenzbedingungen definiert die kohärenten Zustände der Datenbasis. Es existieren im Minimum zwei strukturelle Konsistenzbedingungen, die dem relationalen Datenmodell inhärent sind: Die erste ist die Eindeutigkeit des Schlüssels, welche aussagt, dass eines oder mehrere Attribute den Schlüssel einer Relation darstellen. Das heisst, durch die Wertvergabe eines oder mehrerer Attribute wird ein Tupel eindeutig gekennzeichnet. Die zweite Bedingung ist die referentielle Integrität, welche sicherstellt, dass bestimmte Datenwerte in mehreren Relationen auftreten.

nungen und Anordnung von Zügen im Rahmen der Produktionsplanung der Unternehmung. Das Erstellen des Jahresfahrplans für bestimmte Fahrplanperioden, des Tagesfahrplans (spezielle Anordnungen, Ausfälle, Streckensperrungen usw.) sowie die Ausarbeitung von Fahrplanvarianten sind die Haupttätigkeitsbereiche des Systems.

Die Verkehrsdaten können grafisch oder numerisch auf vordefinierten Strecken eingegeben werden. Dagegen werden die übrigen Daten immer numerisch eingegeben. Ebenfalls numerisch erfolgen die abschliessenden Konstruktionsanpassungen und Ergänzungen sowie die Übergabe der Konstruktionselemente an die Da-

Der betriebswirtschaftliche Aspekt einer möglichst rationellen Bewirtschaftung der Datenressourcen tritt daneben eher in den Hintergrund. Zwar wurde die Struktur der Datenbank Syfa sehr systematisch hergeleitet, aber eben aus der Sicht einer Insellsolution, das heisst erst ab der Stufe Zugriffsmodell (Bild 1).

Im nebenstehenden Kasten sind einige der Relationen in gekürzter Form dargestellt, die im Laufe des Datenentwurfs für Syfa hergeleitet wurden.

Das für Syfa gewählte Vorgehen sieht (jetzt noch) von der Tatsache ab, dass der Fahrplan das Leistungsangebot der SBB schlechthin darstellt und

Einige Relationen des Systems Syfa

- Betriebspunkt (**Betriebspunkt-NR**, Betriebspunkt-Name, Verwaltung, Betriebspunkt-ART...) Anzahl Tupel (Tabellenzellen): ca. 1500
- Abschnitt (**von Betriebspunkt-NR, nach Betriebspunkt-NR...**) Anzahl Tupel: ca. 2000
- Zug (**Zug-NR...**) Anzahl Tupel: ca. 80 000
- Verkehr (**Zug-Nr, von Betriebspunkt-NR, nach Betriebspunkt-NR, Zeit-Abschnitt-Einfahrt...**) Anzahl Tupel: ca. 1 300 000

N.B. Die halbfett gedruckten Attribute bilden die Relationschlüssel. Die Relationen sind hinsichtlich der Attributketten unvollständig

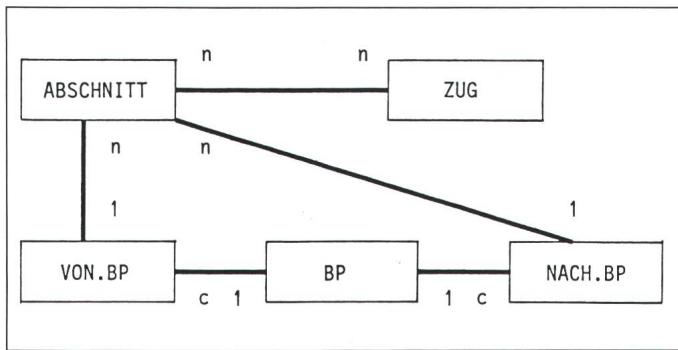


Bild 2
Ausschnitt
Entity-Relationship-
Diagramm Syfa

steme. Die, vielleicht nur vorläufige, Grenze muss gezogen werden zwischen Systemen, welche Fakten² verwalten, und solchen, die wissensbasiert sind oder komplexe Objekte verwalten, welche sich nicht relational abbilden lassen.

Zwar ist die Anwendbarkeit des relationalen Datenmodells auch für die Repräsentation von Wissen grundsätzlich gegeben. (Die im relationalen Datenmodell unterstützten Konsistenzbedingungen können bedingt als Wissen betrachtet werden.) Wissen, relational repräsentiert, führt aber zu Darstellungen, die sehr viele Relationen mit jeweils wenigen Datensätzen (Ausprägungen, Instanzen) enthalten. Auf Fakten basierende Systeme benötigen im Gegensatz dazu Datenbasen mit relativ wenigen Relationen, die viele Ausprägungen enthalten. In diesem Bereich ist aber gerade eine grosse Rationalisierung bei der Bewirtschaftung der Datenressourcen durch eine integrierte Datenhaltung zu erzielen.

Bis jetzt bestehen innerhalb der Systementwicklung bei den SBB praktikable Rezepte zur Datenintegration erst im Bereich der faktenverwaltenden Systeme. In diesem Bereich wird die Integration konsequent durchgeführt, was allmählich zu einer grösseren Übersichtlichkeit der Datenbestände und zu einer wachsenden Klarheit bei unternehmensweiten Begriffen führt.

Untersuchungen, wie sogenannte Non-Standard-Datenbanken, d.h. nicht auf relationale Datenbanken abbildbare Datenbanken, im technischen Bereich in die Integrationsbemühungen einbezogen werden könnten, müssen in nächster Zeit durchgeführt werden. Solche Datenbanken werden demnächst vor allem im Bereich Bau eingesetzt werden, wo die Verwaltung von komplexen Objekten (Pläne, Streckennetz usw.) durch das System «Datenbank der festen Anlagen» automatisiert werden soll.

²«Hans ist Informatiker» stellt ein Faktum dar, während die Aussage «alle Informatiker sind reich» Wissen ist.

schreibung der Datenbasis in möglicherweise verschiedenen Datenbank-Management-Systemen gemacht werden können. Das Vorgehensmodell, welches für die Entwicklung von Informationssystemen bei den SBB verbindlich ist, garantiert die Umsetzung von Integritätsbedingungen, welche auf der Ebene der Entity-Relationship-Diagramme gemacht wurden, auf

Regeln, die zur System-Laufzeit beachtet werden müssen.

Grenzen der Datenintegration

Eine Einschränkung des Geltungsbereiches für die Integration von Datenständen ergibt sich nicht vom Anwendungsbereich der betrachteten Sy-

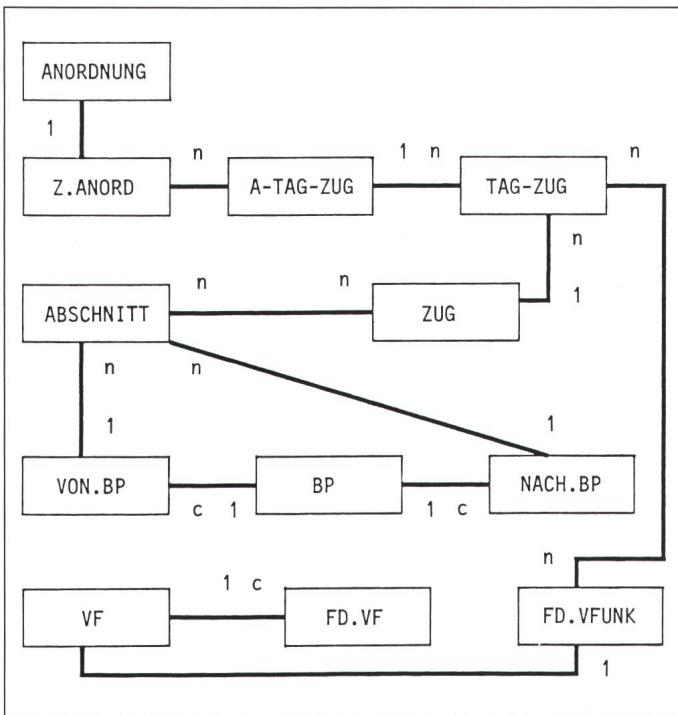


Bild 3
Ausschnitt
Entity-Relationship-
Diagramm Syfa nach
Integration mit dem
System «Betriebliche
Verständigung»