



Kleines Lexikon
der
Informatik
und
Wirtschaftsinformatik

Herausgegeben
von
Universitätsprofessor
Dr. Dr. h. c. Miklós Géza Zilahi-Szabó

R. Oldenbourg Verlag München Wien

Redaktion, textliche und graphische Gestaltung
Dr. Stefan Groos
Dipl.oec.troph. Peter Muthmann

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Kleines Lexikon der Informatik und Wirtschaftsinformatik /

hrsg. von Miklós Géza Zilahi-Szabó. - München ; Wien :

Oldenbourg, 1995

ISBN 3-486-22907-9

NE: Zilahi-Szabó, Miklós G. [Hrsg.]

© 1995 R. Oldenbourg Verlag GmbH, München

Das Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Bearbeitung in elektronischen Systemen.

Gesamtherstellung: R. Oldenbourg Graphische Betriebe GmbH, München

ISBN 3-486-22907-9

INHALTSVERZEICHNIS

Vorwort	VII
Verzeichnis der Autoren	IX
Verzeichnis der Beiträge	XV
Lexikon A - Z	1

VORWORT

Dieses Lexikon soll ein Hilfsmittel für Studenten in ihrem Studium und für an der Informatik Interessierte in ihrer Arbeit sein. Diesem Anspruch entsprechend zeichnet das Lexikon Erscheinungen und Entwicklungen auf dem Gebiet der Informatik, insbesondere der Wirtschaftsinformatik, auf. Es sucht mit Absicht einen Mittelweg zwischen einem Nachschlagewerk über Begriffe und Fakten und einem Handbuch, das Übersichtsdarstellungen liefert. Die Auswahl der Stichworte orientiert sich am aktuellen Standardwissen und an den bedeutsamen Entwicklungen dieser Disziplin. Die Beiträge werden von namhaften Wissenschaftlern und Fachexperten verfaßt. Zur Beschränkung des Umfangs mußte naturgemäß auf eine Reihe von Begriffen verzichtet werden, wobei die große Auswahl der Stichworte (über 180) und der Begriffe bzw. Kurzstichworte (über 500) eine breite Streuung gewährleisten.

Herausgeber und Autoren haben versucht, unter Beachtung der Breite des Benutzerkreises die Stichworte inhaltlich übersichtlich und knapp zu halten - versehen mit vertiefenden Literaturquellen. Die Absicht, an den Anfang der Stichworte jeweils eine Definition zu stellen und die Stichworte nach einer allgemeinen Gliederung inhaltlich abzuhandeln, ließ sich nicht durchgehend verwirklichen.

Zu danken hat der Herausgeber den Verfassern der Stichwortaufsätze, den Mitarbeitern Dr. Stefan Groos und Dipl.oec.troph. Peter Muthmann für die redaktionelle Gestaltung und dem Oldenbourg Verlag, ohne deren Mitwirkung und Unterstützung dieses Lexikon nicht hätte erscheinen können.

Miklós Géza Zilahi-Szabó

VERZEICHNIS DER AUTOREN

Prof. Dr. Heimo H. **Adelsberger**,
Gesamthochschule Essen

Prof. Dr. Dieter **Ahlert**,
Universität Münster

Dr. Werner **Aufsattler**,
Universität Mannheim

Prof. Dr. Heide **Balzert**,
Fachhochschule Dortmund

Dipl.-Inf. V. **Barent**,
Universität Hohenheim

Prof. Dr. Jörg **Becker**,
Universität Münster

Dipl.-Wi.-Inf. Ivonne **Behle**,
Universität Saarbrücken

Prof. Dr. Albrecht **Beutelspacher**,
Universität Gießen

Prof. Dr. Jörg **Biethahn**,
Universität Göttingen

Prof. Dr. Rainer **Bischoff**,
Fachhochschule Furtwangen

Prof. Dr. Arndt **Bode**,
Technische Universität München

Prof. Dr. Freimut **Bodendorf**,
Universität Erlangen-Nürnberg

Dr.-Ing. Gunter **Bolch**,
Universität Erlangen-Nürnberg

Prof. Dr. Hinrich E. G. **Bonin**,
Fachhochschule Lüneburg

Prof. Dr. Walter. **Brenner**,
Technische Universität, Freiberg

Prof. Dr.-Ing. Dr.h.c. Hans-Jörg
Bullinger,
Fraunhofer-Institut Stuttgart

Prof. Dr. Hans **Czap**,
Universität Trier

Prof. Dr. Dr. Ulrich **Derigs**,
Universität Köln

Prof. Dr. Werner **Dirlewanger**,
Universität Kassel

Prof. Dr. Klaus R. **Dittrich**,
Universität Zürich

Prof. Dr. Leonhard **von Dobschütz**,
Universität Istanbul

Prof. Dr.-Ing. Frank-Dieter **Dorloff**,
Gesamthochschule Essen

Dr. Heiner **Dunckel**,
Technische Universität Berlin

Prof. Dr. Günter **Ebert**,
Fachhochschule Nürtingen

Prof. Dr. Dieter **Ehrenberg**,
Universität Leipzig

Prof. Dr.-Ing. K. **Ehrlenspiel**,
Technische Universität München

Prof. Dr. Helmut **Eirund**,
Fachhochschule Harz Wernigerode

Prof. Dr. José **Encarnacao**, et al,
Fraunhofer-Institut Darmstadt

Prof. Dr. Albert **Endres**,
Technische Universität München

Prof. Dr. Hartmut **Ernst**,
Fachhochschule Rosenheim

Verzeichnis der Autoren

Prof. Dr. Werner **Esswein**,
Technische Universität Dresden

Prof. Dr. Günter **Fandel**,
Fernuniversität Hagen

D. **Fischer**,
Universität Göttingen

Prof. Dr. Joachim **Fischer**,
Gesamthochschule Paderborn

Prof. Dr. Roland **Gabriel**,
Universität Bochum

Prof. Dr. Klaus **Garbe**,
Fachhochschule Detmold

Prof. Dr.-Ing. Jürgen **Gausemeier**,
Gesamthochschule Paderborn

Prof. Dr. Kurt **Geihs**,
Universität Frankfurt

Prof. Dr. Hans **Gipper**,
Fachhochschule Aachen

Prof. Dr.-Ing. Winfried **Görke**,
Universität Karlsruhe

Prof. Dr. Günther **Görz**,
Universität Erlangen-Nürnberg

Prof. Dr.-Ing. Dr.h.c. Hans **Grabowski**,
Universität Karlsruhe

Prof. Dr. Martin **Graef**,
Württembergische
Gebäudebrandversicherung Stuttgart

Prof. Dr. Manfred **Grauer**,
Gesamthochschule Siegen

Dipl.-Ing. Gerhard **Greiner**,
Joanneum Research
Forschungsgesellschaft mbH Graz

Prof. Dr. Joachim **Griese**,
Universität Bern

Dr. Hans-Dieter **Groffmann**,
Universität Tübingen

Dipl.-Inf. Thomas **Grotehen**,
Universität Zürich

Prof. Dr. Hans-Otto **Günther**,
Technische Universität Berlin

Prof. Dr. Rul **Gunzenhäuser**,
Universität Stuttgart

Prof. Dr. Hans-Dietrich **Haasis**,
Universität Bremen

Prof. Dr. Klaus **Haefner**,
Universität Bremen

Prof. Dr. Udo **Hahn**,
Universität Freiburg

Prof. Dr. Winfried **Hassemer**,
Der Hessische Datenschutzbeauftragte
Wiesbaden

Dipl.-Betriebsw. BA Jürgen **Hausin**,
Berufsakademie Lörrach

Prof. Dr. Wolfgang **Hesse**,
Universität Marburg

Prof. Dr. Andreas **Heuer**,
Universität Rostock

Dr. Lorenz M. **Hilty**,
Universität Hamburg

Prof. Dr.-Ing. Hans-Jürgen **Hoffmann**,
Technische Hochschule Darmstadt

Prof. Dr. Fridolin **Hofmann**,
Universität Erlangen-Nürnberg

Prof. Dr. Christian **Horn**,
Fachhochschule Furtwangen

Prof. Dr. Josef **Hoschek**,
Technische Hochschule Darmstadt

Verzeichnis der Autoren

Prof. Dr. Uwe **Hübner**,
Technische Universität Chemnitz-
Zwickau

Prof. Dr. Wilhelm **Hummeltenberg**,
Universität Hamburg

Prof. Dr. Dipl.-Inf. J. Anton **Illik**,
Fachhochschule Furtwangen

lic.rer.pol. Beat **Jaccottet**,
Universität Bern

Prof. Dr. Kai **Jacobs**,
Technische Hochschule Aachen

Prof. Dr. Bernd **Jahnke**,
Universität Tübingen

Prof. Dr. Wolfgang **Janko**,
Wirtschaftsuniversität Wien

Prof. Dr. Gerhard R. **Joubert**,
Technische Universität Clausthal-
Zellerfeld

Prof. Dr. Werner **Junginger**,
Universität der Bundeswehr Hamburg

Prof. Dr. Herbert **Kargl**,
Universität, Mainz

Dipl.-Wirtsch.-Ing. Rainer A. **Kellerhals**,
Universität Kaiserslautern

Prof. Dr. Michael **Kerres**,
Fachhochschule Furtwangen

Dr.-Ing. Alfons **Kiewert**,
Technische Universität München

Prof. Dr. Herbert **Klaeren**,
Universität Tübingen

Dr. Wolfgang **Klas**,
Technische Hochschule Darmstadt

Prof. Dr. Gerhard **Knolmayer**,
Universität Bern

Prof. Dr.-Ing. Gerhard **Knorz**,
Fachhochschule Darmstadt

Dr. **Koch**,
Scientific Consulting Köln

Prof. Dr. Lothar **Koch**,
Fachhochschule Dresden

Prof. Dr. Rainer **Koch**,
Gesamthochschule Paderborn

Prof. Dr. Wolfgang **Köhler**,
Universität Gießen

Prof. Dr. Gerhard **Kongehl**,
Universität Ulm

Prof. Dr. Hermann **Krallmann**,
Technische Universität Berlin

Prof. Dr. Helmut **Krcmar**,
Universität Hohenheim

Prof. Dr. R. **Krieger**,
Universität Karlsruhe

Prof. Dr. Rudolf **Kruse**,
Technische Universität Braunschweig

Prof. Dr. Ingbert **Kupka**,
Technische Universität, Clausthal-
Zellerfeld

Prof. Dr.-Ing. Hermann Werner **Kurth**,
Fachhochschule Gießen-Friedberg

Prof. Dr. Horst **Langendörfer**,
Technische Universität Braunschweig

Prof. Dr. Georg **Lausen**,
Universität Freiburg

Prof. Dr. Hermann **Locarek-Junge**,
Gesamthochschule Essen

Prof. Dr. Rita **Loogen**,
Universität Marburg

Verzeichnis der Autoren

Dr. Michael **Löwe**,
Fraunhofer-Einrichtung, Dortmund

Prof. Dr. Jochen **Ludewig**,
Universität Stuttgart

Dipl.-Ing. E. **Ludwig**,
Universität Hannover

Prof. Dr. Markus **Lusti**,
Universität Basel

Prof. Dr. Bernd **Mahr**,
Technische Universität Berlin

Dr.- Ing. Kai **Mertins**,
Technische Universität Berlin

Prof. Dr. Hans-Werner **Meuer**,
Universität Mannheim

Dr. Frank **Morelli**,
Universität Stuttgart

Prof. Dr. Wilhelm **Mülder**,
Fachhochschule Mönchengladbach

Prof. Dr. Johann-Adolf **Müller**,
Fachhochschule Dresden

Prof. Dr. Günter **Müller**,
Universität Freiburg

Prof. Dr. Heiner **Müller-Merbach**,
Universität Kaiserslautern

Prof. Dr.-Ing. Christian **Müller-Schloer**,
Universität Hannover

Prof. Dr. Peter **Naeve**,
Universität Bielefeld

Prof. Dr.-Ing. M. **Nagl**,
Technische Hochschule Aachen

Prof. Dr. Erich J. **Neuhold**,
Technische Hochschule Darmstadt

Prof. Dr. Gerhard **Niemeyer**,
Universität Regensburg

Dr. A. **Oberweis**,
Universität Karlsruhe

Dr. Andreas **Oestreicher**,
Universität Mannheim

Dr. Rainer **Olbrich**,
Universität Münster

Prof. Dr. Erich **Ortner**,
Universität Konstanz

Prof. Dr. Uwe **Pape**,
Technische Universität, Berlin

Prof. Dr. **Peschanel**,
Technische Universität Ilmenau

Prof. Dr.-Ing. Ilka **Philippow**,
Technische Universität Ilmenau

Prof. Dr. Dieter B. **Preßmar**,
Universität Hamburg

Prof. Dr. Frank **Puppe**,
Universität Würzburg

Dr. Thomas **Rauber**,
Universität Saarbrücken

Prof. Dr. Wolf **Rauch**,
Universität Graz

Prof. Dr. Otto **Rauh**,
Fachhochschule Heilbronn Künzelsau

Dr. Claus **Rautenstrauch**,
Universität Münster

Dipl.-Ing. Gerd-Joachim **Reefmann**,
Universität Hannover

Prof. Dr. Heinz **Rehkugler**,
Universität Freiburg

Verzeichnis der Autoren

Prof. Dr. Michael **Reiß**,
Universität Stuttgart

Prof. Dr. Lutz **Richter**,
Universität Zürich

Dr. Eike Hagen **Riedemann**,
Universität Dortmund

Prof. Dr. Johannes **Röhrich**,
Universität Jena

Prof. Dr.-Ing. Karlheinz **Roschmann**,
Fachhochschule Konstanz

Dr. Michael **Salmony**
IBM - Europäisches Zentrum für
Netzwerkforschung
Heidelberg

Prof. Dr. August-Wilhelm **Scheer**,
Universität Saarbrücken

Prof. Dr. Heinz **Schelle**,
Universität der Bundeswehr München

Prof. Dr. Jürgen **Scherff**,
Fachhochschule Furtwangen

Prof. Dr. Bernd **Schiemenz**,
Universität Marburg

Prof. Dr. Henner **Schierenbeck**,
Universität Basel

Prof. Dr. Alexander **Schill**,
Technische Universität Dresden

Prof. Dr. Reinhard **Schramm**,
Technische Universität Ilmenau

Prof. Dr. Herbert **Schreiter**,
Technische Universität Chemnitz-
Zwickau

Dr. **Schulte-Hillen**,
Scientific Consulting Köln

Prof. Dr.-Ing. Manfred **Schulte-
Zurhausen**,
Fachhochschule Aachen

Prof. Dr. Peter **Schulthess**,
Universität Ulm

Dipl.-Kfm. Hermann **Schuster**,
Universität Stuttgart

em. Prof. Dr. Gunter **Schwarze**,
Humboldt-Universität Berlin

Prof. Dr. Jochen **Schwarze**,
Technische Universität Hannover

Prof. Dr. Manfred **Seifert**,
Fachhochschule Karlsruhe

Dr. Werner **Sinzig**,
SAP AG, Walldorf

Prof. Dr. Peter **Sokolowsky**,
European Business School, Oestrich-
Winkel

Prof. Dr. Otto **Spaniol**,
Technische Hochschule Aachen

Prof. Dr. Jürgen **Spielmann**,
Fachhochschule Würzburg-Schweinfurt

Prof. Dr.h.c.mult Dr.- Ing. Günter **Spur**,
Technische Universität Berlin

Prof. Dr. Peter **Stahlknecht**,
Universität Osnabrück

Prof. Dr. Franz **Steffens**,
Universität Mannheim

Dr.-Ing. Wolfgang **Stein**,
DeTeMobil GmbH Münster

Prof. Dr. Werner **Steinmüller**,
Universität Bremen

Prof. Dr. H. **Stever**,
Gesamthochschule Paderborn

Verzeichnis der Autoren

Prof. Dr. Eberhard **Stickel**,
Universität Frankfurt/Oder

Dr. Ralf **Strauß**,
Universität Freiburg

Prof. Dr. Franz **Stuchlik**,
Universität Magdeburg

Prof. Dr. Wolffried **Stucky**,
Universität Karlsruhe

Prof. Dr. Uwe **Suhl**,
Freie Universität, Berlin

Prof. Dr. Djamshid **Tavangarian**,
Fern-Universität Hagen

Prof. Dr. Roland **Traummüller**,
Universität Linz

Prof. Dr. Jürgen **Treffert**,
Berufsakademie Lörrach

Prof. Dr. J. Klaus G. **Troitsch**,
Universität Koblenz-Landau

Prof. Dr. Wolfgang **Ülzmann**,
Fachhochschule Wedel

Prof. Dr. Wolfgang **Uhr**,
Technische Universität Dresden

Prof. Dr. Theo **Ungerer**,
Universität Karlsruhe

Prof. Dr. Richard **Vahrenkamp**,
Gesamthochschule Kassel

Prof. Dr. Rudolf **Vetschera**,
Universität Konstanz

PD Dr. Max **Vetter**,
Eidgen. Technische Hochschule Zürich

Dipl.-Ing. Dozent Stefan **Vollmer**,
Dr. Hartmann & Partner AG
Lörrach/Winterthur

Prof. Dr. Reinhard **Voßbein**,
Gesamthochschule Essen

Prof. Dr. Gottfried **Vossen**,
Universität Münster

Prof. Dr. Herbert **Weber**,
Fraunhofer-Einrichtung, Dortmund

Prof. Dr.-Ing. E.h. Hans-Peter **Wiendahl**,
Universität Hannover

Dipl.-Kfm. Rüdiger **Wilbert**,
Universität Trier

Prof. Dr.-Ing. Horst **Wildemann**,
Technische Universität München

Prof. Dr. Klaus Werner **Wirtz**,
Fachhochschule Mönchengladbach

Prof. Dr. Frank-Jürgen **Witt**,
Transfer-Institut Management
Controlling Kenzingen

Prof. Dr. M.-R. **Wolff**,
Gesamthochschule Wuppertal

Dipl.-Stat. Bernhard **Wunder**,
Universität Koblenz-Landau

Prof. Dr. Hans Christoph **Zeidler**,
Universität der Bundeswehr Hamburg

Prof. Dr. Dr.h.c. Miklós Géza **Zilahi-Szabó**
Justus-Liebig-Universität Giessen

Prof. Dr. Dr.h.c. Hans-Jürgen
Zimmermann,
Technische Hochschule Aachen

Prof. Dr. Heinz **Züllighoven**,
Universität Hamburg

Dr. Volker **Zurwehn**,
Fraunhofer Einrichtung Dortmund

VERZEICHNIS DER BEITRÄGE

- Abfragesprachen, relationale**
(Prof. Dr. W. Stucky, Dr. A. Oberweis, Karlsruhe)
- Analysemethoden, objektorientierte**
(Prof. Dr. H. Balzert, Dr.-Ing. W. Stein, Dortmund)
- Anwendungssoftware für das Rechnungswesen**
(Dr. W. Sinzig, Walldorf)
- Anwendungssysteme**
(Prof. Dr. R. Gabriel, Bochum)
- Anwendungssysteme in Dienstleistungsbetrieben**
(Prof. Dr. Peschanel, Ilmenau)
- Aus- und Weiterbildungssysteme, computergestützte**
(Prof. Dr. F. Bodendorf, Nürnberg)
- Ausbildung in der Wirtschaftsinformatik an Hochschulen**
(Prof. Dr. R. Gabriel, Bochum)
- Ausbildung von Ingenieurstudenten in der Informatik**
(Prof. Dr. H. Schreiter, Chemnitz-Zwickau)
- Auswahlprozeß von Hardware, Software und Dienstleistungen**
(Prof. Dr. R. Bischoff, Furtwangen)
- Bank-Controlling**
(Prof. Dr. H. Schierenbeck, Basel)
- BDE (Betriebsdatenerfassung)**
(Prof. Dr. K. Roschmann, Konstanz)
- Benutzungsoberflächen, graphische**
(Prof. Dr. J. Röhrich, Jena)
- Berufsfelder der Wirtschaftsinformatik**
(Prof. Dr. H. Gipper, Aachen)
- Betriebsmodelle**
(Prof. Dr. F. Steffens, Mannheim)
- Betriebssystem**
(Prof. Dr. F. Hofmann, Erlangen-Nürnberg)
- Bilddatenverarbeitung**
(Prof. Dr. H. Ernst, Rosenheim)
- Bildschirmtext (Btx), Videotex (VTX), Datex-J (DxJ), Public Access Network (PAN)**
(Dipl.-Ing. G. Greiner, Graz)
- Bildungsinformatik**
(Prof. Dr. H. Stever, Dipl.-Stat. B. Wunder, Paderborn)
- Biometrie und Datenverarbeitung**
(Prof. Dr. W. Köhler, Gießen)
- Breitbandkommunikation**
(Prof. Dr. U. Pape, Berlin)
- Büro, Aufgabenanalyse im**
(Dr. H. Dunckel, Berlin)
- Bürokommunikation**
(Prof. Dr. H. Krallmann, Berlin)
- CAD-Anwendungen**
(Prof. Dr. H. W. Kurth, Gießen)
- CAD-Systeme**
(Prof. Dr. J. Hoschek, Darmstadt)
- CARE**
(Prof. Dr. H. Weber, M. Löwe, Berlin)
- CASE**
(Dr. V. Zurwehn, Dortmund)
- Case Based Reasoning**
(Prof. Dr. D. Ehrenberg, Leipzig)
- CBT (Computerbasiertes Training)**
(Prof. Dr. M. Kerres, Furtwangen)
- CIB (Computer Integrated Business)**
(Prof. Dr. Dr.h.c. H.-J. Bullinger, Stuttgart)
- CIM (Computer Integrated Manufacturing)**
(Prof. Dr. A.-W. Scheer, Saarbrücken)
- Client/Server-Architekturen**
(Dr. W. Aufsattler, Mannheim)
- Client/Server-Informationssysteme**
(Prof. Dr. J. Treffert, Dipl.-Betriebswirt J. Hausin, Lörrach)
- Computergraphik**
(Prof. Dr. T. Rauber, Saarbrücken)
- Computerlinguistik**
(Prof. Dr. U. Hahn, Freiburg)

Verzeichnis der Beiträge

Controlling, DV-

(Prof. Dr. H. Kargl, Mainz)

Controlling, Informationssysteme im

(Prof. Dr. G. Ebert, Nürtingen)

Corporate Network

(Prof. Dr. H. Locarek-Junge, Essen)

CSCW im Kommunalbereich

(Prof. Dr. R. Traunmüller, Linz)

Datenbanken, online-

(Dr. Schulte-Hillen, Dr. Koch, Köln)

Datenbanksysteme

(Prof. Dr. R. Krieger, Prof. Dr. W. Stucky, Karlsruhe)

Datenbanksysteme im Büro

(Prof. Dr. M.-R. Wolff, Wuppertal)

Datenbanksystemen, Architektur von

(Prof. Dr. G. Vossen, Münster)

Datenmanagement im ganzheitlichen

Informationsmanagement

(Prof. Dr. J. Biethahn, D. Fischer, Göttingen)

Datenmodellierung

(PD Dr. M. Vetter, Zürich)

Datenschutz

(Prof. Dr. W. Hassemer, Frankfurt)

Datenschutz in der Datenverarbeitung

(Prof. Dr. G. Kongehl, Ulm)

Datensicherung

(Prof. Dr. J. A. Illik, Furtwangen)

Design, kostengesteuertes

(Dr.-Ing. A. Kiewert, München)

Dispositionssysteme

(Prof. Dr. B. Schiemenz, Marburg)

Dokumentenverarbeitung

(Prof. Dr. R. Gabriel, Bochum)

Dokumentenverarbeitung in Büroprozessen, Integrierte

(Prof. Dr. R. Voßbein, Essen)

Downsizing

(Prof. Dr. G. Knolmayer, B. Jaccottet, Bern)

DSS (Decision Support System)

(Prof. Dr. W. Hummeltenberg, Hamburg)

DV-Leistung, anwendungsorientierte Messung und Bewertung der

(Prof. Dr. W. Dirlwanger, Kassel)

EIS

(Prof. Dr. H. Krcmar, V. Barent, Hohenheim)

Entwicklungsmanagement

(Prof. Dr. H. W. Kurth, Gießen)

Expertensysteme in der Wirtschaft

(Prof. Dr. I. Kupka, Clausthal-Zellerfeld)

Expertensystemen,

Problemlösungsmethoden in

(Prof. Dr. F. Puppe, Würzburg)

Fertigungsablaufdiagnose

(Prof. Dr. H.-P. Wiendahl, Dipl.-Ing. E. Ludwig, Hannover)

Flynnsche Klassifizierung

(Prof. Dr. Th. Ungerer, Karlsruhe)

Führungsinformationssystem

(Dr. H.-D. Groffmann, Tübingen)

Fuzzy-Systeme

(Prof. Dr. R. Kruse, Braunschweig)

Fuzzy-Technologie

(Prof. Dr. Dr.h.c. H. J. Zimmermann, Aachen)

GDSS

(Group Decision Support Systems)

(Prof. Dr. R. Vetschera, Konstanz)

Geschäftsprozesse und Geschäftsprozessanalysen

(Prof. Dr. W. Esswein, Dresden)

Handelsinformationssysteme

(Prof. Dr. J. Becker, Münster)

Hypermedia

(A. Kellerhals, Kaiserslautern)

Individuellen Datenverarbeitung, Gestaltung der

(Prof. Dr. O. Rauh, Heibronn-Künzelsau)

Informatik, Angewandte

(Prof. Dr. W. Steinmüller, Bremen)

Informatik, die Zukunft der (Wirtschafts-)

(Prof. Dr. K. Haefner, Bremen)

**Information Highways
/Datenautobahnen**

(Dr. Salmony, M. Heidelberg)

Information Retrieval

(Prof. Dr. W. Rauch, Graz)

Information Retrieval-Anwendungen

(Prof. Dr.-Ing. G. Knorz, Darmstadt)

Information Retrieval-Systeme

(Prof. Dr. R. Schramm, Ilmenau)

Information und Gesellschaft

(Prof. Dr. K. G. Troitzsch, Koblenz)

**Information und Kommunikation,
Management von**

(Prof. Dr. H. Müller-Merbach, Kaiserslautern)

**Informations- und Kommunikations-
technik, Planung des Einsatzes von**

(Prof. Dr. E. Stickle, Frankfurt/Oder)

Informationsflußintegration

(Prof. Dr. H. Wildemann, München)

Informationsmanagement

(Prof. Dr. J. Griese, Bern)

**Informationsmanagement,
Kooperation im**

(Dipl.-Kfm. H. Schuster, Stuttgart)

**Informationsmanagement,
multimediales**

(Dr. W. Klas, Prof. Dr. E. J. Neuhold,
Darmstadt)

**Informationsmanagement,
strategisches**

(Prof. Dr. W. Brenner, Freiberg)

Informationssysteme, multimediale

(Prof. Dr. M.-R. Wolff, Wuppertal)

Informationssysteme, verteilte

(Prof. Dr. H. Locarek-Junge, Essen)

Informationssystemen, Planung von

(Prof. Dr. W. Esswein, Dresden)

**Informationsverarbeitung in den
betrieblichen Funktionsbereichen**

(Prof. Dr. P. Stahlknecht, Osnabrück)

**Informationsverarbeitung in der
Logistik, integrierte**

(Prof. Dr. F.-D. Dorloff, Essen)

**Informationsverarbeitung in der
Produktion, integrierte**

(Prof. Dr. H. H. Adelsberger, Essen)

**Informationswert,
Informationsbewertung**

(Prof. Dr. W. H. Janko, Wien)

Jahresabschlußanalyse, dv-gestützte

(Dr. A. Oestreicher, Mannheim)

JIT (Just in Time)

(Prof. Dr. H. Wildemann, München)

Kommunikation, zwischenbetriebliche

(Prof. Dr. M. Schulte-Zurhausen,
Aachen)

**Kommunikationsstrukturen in
heterogenen verteilten Systemen**

(Prof. Dr. K. Geihs, Frankfurt)

Kommunikationssysteme, Offene

(Prof. Dr. O. Spaniol, K. Jakobs,
Aachen)

**Konstruieren und Planen,
rechnerunterstütztes**

(Prof. Dr. R. Koch, Paderborn)

Kooperationskultur im EDV-Bereich

(Prof. Dr. M. Reiß, Dr. F. Morelli,
Stuttgart)

Kryptologie

(Prof. Dr. A. Beutelspacher, Gießen)

**Lagerhaltung und Materialdisposition
mittels Datenverarbeitung**

(Prof. Dr. H.-O. Günther, Berlin)

Leistungsbewertung

(Dr.-Ing. G. Bolch, Erlangen)

Leitstand

(Dr. K. Mertins, Prof. Dr. G. Spur,
Berlin)

Leitsysteme in der Fertigung

(Prof. Dr. J. Gausemeier, Paderborn)

Marketing-/Vertriebsinformationssystem

(Prof. Dr. J. Treffert, Lörrach; Dipl.-
Ing. St. Vollmer, Winterthur)

Medienmanagement

(Prof. Dr. G. Müller, Dr. R. Strauß,
Freiburg)

**medizinischen Anwendung, Software-
technik in der**

(Prof. Dr. B. Mahr, Berlin)

Metadatenmodelle

(Prof. Dr. H. Czap, Trier)

Verzeichnis der Beiträge

Metainformationssysteme, Architektur betrieblicher

(Prof. Dr. E. Ortner, Konstanz)

Modellierung und Simulation

(em. Prof. Dr. G. Schwarze, Berlin)

MSS (Management Support Systems)

(Prof. Dr. W. Uhr, Dresden)

Multimediasysteme

(Prof. Dr. H. Eirund, Wernigerode)

Multimediasysteme in der

Wirtschaftspraxis

(Prof. Dr. G. R. Joubert,
Clausthal-Zellerfeld)

Multimediawelt, Datenbanken in der

(Prof. Dr. P. Sokolowsky,
Oestrich-Winkel)

Multiprozessorsysteme

(Prof. Dr. Th. Ungerer, Karlsruhe)

Netzdienste, elektronische

(Prof. Dr.-Ing. Ch. Müller-Schloer, G.
Reefmann, Hannover)

Neuronale Netze

(Prof. Dr. H. Rehkugler, Freiburg)

Objektmodellierung

(Prof. Dr. K. R. Dittrich, Dipl.-Inform.
Thomas Grotehen, Zürich)

Objektorientierte Methoden für Analyse, Entwurf und Programmierung

(Prof. Dr. H.-J. Hoffmann, Darmstadt)

Offene Systeme, Standards für

(Prof. Dr. J. A. Illik, Furtwangen)

Offener Systeme, Architektur

(Prof. Dr. M. Seifert, Karlsruhe)

Operations Research und

Wirtschaftsinformatik

(Prof. Dr. H. Müller-Merbach,
Kaiserslautern)

Optimierungsproblemen, parallele und verteilte Lösung von

(Prof. Dr. M. Grauer, Siegen)

Optimierungssoftware für betriebliche Anwendungen

(Prof. Dr. U. Suhl, Berlin)

Organizational Intelligence

(Prof. Dr. H. Müller-Merbach,
Kaiserslautern)

OSF Distributed Computing Environment

(Prof. Dr. A. Schill, Dresden)

Outsourcing

(Prof. Dr. G. Knolmayer, Bern)

Parallelrechnerarchitektur

(Prof. Dr. A. Bode, München)

PC- und Workstationarchitektur

(Prof. Dr. Ch. Horn, Furtwangen)

Personalcomputing

(Prof. Dr. R. Vahrenkamp, Kassel)

Personalinformationssysteme

(Prof. Dr. W. Müller,
Mönchengladbach)

Planungssysteme, computergestützte

(Prof. Dr. D. B. Preßmar, Hamburg)

PPS-Systeme (Produktionsplanungs- und -steuerungssysteme)

(Prof. Dr. G. Fandel, Hagen)

Preis-/Leistungsverhältnis

(Prof. Dr. W. Dirlwanger, Kassel)

Programmierung, objektorientierte

(Prof. Dr. W. Junginger, Hamburg)

Programmierung, parallele

(Prof. Dr. R. Loogen, Marburg)

Projekt- und Prozeßplanung

(Prof. Dr. G. Niemeyer, Regensburg)

Projektmanagement

(Prof. Dr. H. Schelle, München)

Projektpriorisierung

(Prof. Dr. L. von Dobschütz, Istanbul)

Prototyping

(Prof. Dr. H. Züllighoven, Hamburg)

Prozeßautomatisierung/-datenverarbeitung

(Dr. G. Bolch, Erlangen)

Prozeßinformatik

(Prof. Dr.-Ing. I. Philippow, Ilmenau)

Publizieren, computergestütztes

(Prof. Dr. U. Pape, Berlin)

Quantitative Methoden bei der Planung und Entwicklung von Informationssystemen

(Prof. Dr. J. Schwarze, Hannover)

Re-Engineering

(Prof. Dr. L. Richter, Zürich)

Rechenzentren, Organisation und Betrieb von

(Prof. Dr. M. Graef, Stuttgart)

Rechnerkommunikation über ISDN

(Prof. Dr. P. Schulthess, Ulm)

Rechnernetze

(Prof. Dr. U. Hübner, Chemnitz)

Rechnernetzen, Management von

(Prof. Dr. K. Garbe, Detmold)

Rechtsinformatik

(Prof. Dr. W. Steinmüller, Bremen)

Relationenmodell

(Prof. Dr. G. Lausen, Freiburg)

Robotersysteme

(Prof. Dr. W. Ülzmann, Wedel)

Sicherheit in der Informationstechnik

(Prof. Dr. H. Langendörfer,
Braunschweig)

Simultaneous Engineering

(Prof. Dr. Dr. h.c. H. Grabowski,
Karlsruhe)

Software Engineering

(Prof. Dr. H.-J. Spielmann, Würzburg)

Software-Entwicklungs-umgebungen

(Prof. Dr.-Ing. M. Nagl, Aachen)

Softwareanwendungssystemen, Entwicklungstendenzen in

(Prof. Dr. K. W. Wirtz,
Mönchengladbach)

Softwareprüfung/-qualitätssicherung

(Prof. Dr. J. Ludewig, Stuttgart)

Softwaretestmethoden

(Dr. E. H. Riedemann, Dortmund)

Softwarewartung

(Prof. Dr. L. Richter, Zürich)

Sprachverarbeitung

(Prof. Dr. G. Görz, Erlangen-Nürnberg)

SQL-Normen

(Prof. Dr. A. Heuer, Rostock)

Statistics, Computational

(Prof. Dr. P. Naeve, Bielefeld)

Strukturierte Programmierung

(Prof. Dr. H. Klaeren, Tübingen)

Symplex-Graphiken

(Prof. Dr. H. Müller-Merbach,
Kaiserslautern)

Systemanalyse

(Prof. Dr. W. Hesse, Marburg)

Systemanalyse, Basistechniken der

(Prof. Dr. H. Balzert, Dortmund)

Systemarchitektur

(Prof. Dr.-Ing. H. Ch. Zeidler,
Hamburg)

Systeme, objektorientierte

(Prof. Dr. H. Eirund, Wernigerode)

Systemforschung

(Prof. Dr.-Ing. J.-A. Müller, Dresden)

Tourenplanung (Tourendisposition)

(Prof. Dr. U. Derigs, Köln)

Tutorielle Systeme, Intelligente

(Prof. Dr. M. Lusti, Basel)

Umweltinformationssysteme, betriebliche

(Dr. L. M. Hilty, Dr. C. Rautenstrauch,
Hamburg, Münster)

Umweltschutzinformationssysteme und Umwelt-PPS, betriebliche

(Prof. Dr. H.-D. Haasis, Bremen)

UNIX

(Prof. Dr. L. Koch, Dresden)

Versicherungsunternehmen, Informations- und Kommunikationstechnik in

(Prof. Dr. J. Scherff, Furtwangen)

Vertriebscontrolling, Softwaredesign für das

(Prof. Dr. F.-J. Witt,
Kenzingen b. Freiburg)

Verwaltung, integrierte Anwendungssysteme in der

(Prof. Dr. H. Locarek-Junge, Essen)

Verwaltungsinformatik

(Prof. Dr. H. E. G. Bonin, Lüneburg)

Virtual Environments/Virtual Reality

(Prof. Dr. J. L. Encarnação, et al.,
Darmstadt)

Visualisierung ökonomischer Daten

(Prof. Dr. B. Jahnke, Tübingen)

Verzeichnis der Beiträge

**Warenwirtschaftsmanagement,
computergestütztes (CWWS)**
(Prof. Dr. D. Ahlert, R. Olbrich,
Münster)

Werbung, computergestützte
(Dipl. Wi.-Inf. Ivonne Behle,
Saarbrücken)

Wirtschaftsgut Software
(Prof. Dr. A. Endres, München)

**Wissensbasierte Systeme in der Bera-
tung**
(Prof. Dr. F. Stuchlik, Magdeburg)

Workstations
(Prof. Dr. D. Tavangarian, Hagen)

**Zuverlässigkeit und Fehlertoleranz von
Rechensystemen**
(Prof. Dr.-Ing. Görke, Karlsruhe)

**Zwischenbetriebliche Integration in
der Informationsverarbeitung**
(Prof. Dr. J. Fischer, Paderborn)

Abfragesprachen, relationale

Relationale Abfragesprachen ermöglichen das Finden von Daten in einer relationalen Datenbank und die Bereitstellung der gesuchten Daten in einer Ergebnisrelation. Dabei gibt es zwei unterschiedliche Vorgehensweisen:

- bei der *prozeduralen Formulierung* einer Datenbankanfrage wird eine Folge von Operationen angegeben, die aus den gegebenen Relationen die Ergebnisrelation erzeugt;
- bei der *deklarativen Formulierung* wird die Ergebnisrelation durch Bedingungen definiert, ohne die genaue Vorgehensweise zur Erzeugung der Ergebnisrelation angeben zu müssen.

Als Beispielrelationen werden eine Angestellte- und Projektrelation betrachtet.

angestellte	ANG-NR	NAME	WOHN-ORT
	3115	Meyer	Karlsruhe
	3207	Müller	Mannheim
	2814	Klein	Mannheim
	3190	Maus	Karlsruhe
	2314	Groß	Karlsruhe
	1324	Schmitt	Heidelberg
	1435	Mann	Bruchsal
	2412	Müller	Karlsruhe
	2244	Schulz	Bruchsal
	1237	Krämer	Ludwigsh.
	3425	Schmitz	Pforzheim
	2454	Schuster	Worms

projekte	P-NR	ANG-NR	PROZ-ARBZEIT
	761235	3207	100
	761235	3115	50
	761235	3190	50
	761235	1435	40
	761235	3425	50
	770008	2244	20
	770008	1237	40
	770008	2814	70
	770008	2454	40
	770114	2814	30

projekte	P-NR	ANG-NR	PROZ-ARBZEIT
	770114	1435	60
	770114	1237	60
	770114	2454	60
	770114	3425	50
	770114	2412	100
	770231	3190	50
	770231	2314	100
	770231	2244	80
	770231	3115	50
	770231	1324	100

Dazu soll folgende Datenbankanfrage formuliert werden: Finde die Namen aller Angestellten aus Karlsruhe, die an Projekt 770231 mitarbeiten, und gib für diese Angestellten an, wie hoch der für das Projekt aufgewendete jeweilige prozentuale Anteil an der Gesamtarbeitszeit ist. Die Ergebnisrelation zu dieser Anfrage ist aus der Tabelle ersichtlich.

NAME	PROZ-ARBZEIT
Meyer	50
Maus	50
Groß	100

Es werden nachfolgend vier wichtige Typen relationaler Abfragesprachen kurz beschrieben:

- Die *Relationenalgebra* ist eine prozedurale Abfragesprache für relationale Datenbanken. Es werden die folgenden Grundoperationen bereitgestellt: Mengenoperationen (Schnittmenge \cap , Vereinigung \cup , kartesisches Produkt \times). *Projektion* $\pi[L]$ rel zur Entfernung der Spalten aus einer Tabelle rel, die nicht in L, einer Menge von Attributnamen aus rel, enthalten sind. Es ist möglich, daß nach Anwendung einer Projektion gleiche Zeilen vorkommen; als Folge werden die überflüssigen Zeilen gestrichen. *Selektion* $\sigma[\text{Sel.-Bed.}]$ rel zur Auswahl von Zeilen einer Tabelle rel, die eine gegebene Selektionsbedingung erfüllen. *Join (Verbund)* zur Verknüpfung zweier Relationen zu einer Relation (möglicherweise) höheren Grades. Die Verknüpfung

kann nach mehreren Kriterien erfolgen, entsprechend wird (u.a.) zwischen Natural Join (natürlicher Verbund) und Theta-Join (Theta-Verbund) unterschieden. Beim Natural Join $rel1 * rel2$ werden zwei Relationen $rel1$ und $rel2$ bzgl. gemeinsamer (Join-) Attribute miteinander verknüpft. Dazu wird jedes Tupel aus der einen Relation mit jedem Tupel aus der anderen Relation, das bzgl. der Join-Attribute gleiche Werte hat, zu einem neuen Ergebnis-Tupel zusammengesetzt. Beim Theta-Join $rel1 * [Join-Bed.] rel2$ werden Relationen $rel1$ und $rel2$ nicht unbedingt bzgl. gleicher Attribute miteinander verbunden, es können auch unterschiedliche Attribute mit einer sog. Join-Bedingung miteinander verglichen werden. Falls die Join-Bedingung für zwei Tupel erfüllt ist, dann werden die beiden Tupel zu einem neuen Tupel der Ergebnisrelation zusammengesetzt.

Die Anwendung einer dieser Operationen auf eine gegebene Relation erzeugt wiederum eine neue Relation. Es können komplexe Ausdrücke gebildet werden, die von rechts nach links bzw. von innen nach außen interpretiert werden. Bei Bedarf kann mit Hilfe der Klammersetzung auch eine andere Abarbeitungsreihenfolge vorgegeben werden.

Die Beispielanfrage lautet in der Relationenalgebra (auch andere äquivalente Formulierungen sind möglich!):

$\pi[NAME, PROZ-ARBZEIT]$
 $(\sigma[WOHNORT="Karlsruhe" \wedge P-NR=770231](angestellte * projekt))$.

Bei der Abarbeitung dieser Anfrage wird zunächst die Angestelltenrelation mit der Projektrelation gejoint. Dann werden die Angestellten aus Karlsruhe, die am Projekt 770231 mitarbeiten, selektiert und die Ergebnisrelation wird schließlich auf die Attribute NAME und PROZ-ARBZEIT projiziert.

Beim (Tupel-) Relationenkalkül wird eine Ergebnisrelation definiert, indem eine logische Bedingung angegeben wird, die die Ergebnisrelation erfüllen muß. Es kann gezeigt werden, daß Relationenkalkül und Relationenalgebra die gleiche Ausdrucksmächtigkeit besitzen. Eine Anfrage im Relationenkalkül hat die Form $\{t \mid P(t)\}$ und bedeutet: "Suche alle Tupel t , für die der logische Ausdruck $P(t)$ wahr

ist". Die Beispielanfrage lautet im (Tupel-) Relationenkalkül:

$\{t \mid \exists u \in angestellte (\exists v \in projekte (u.ANG-NR = v.ANG-NR \wedge u.WOHNORT="Karlsruhe" \wedge t.NAME = u.NAME \wedge t.PROZ-ARBZEIT=v.PROZ-ARBZEIT))\}$.

- \uparrow Structured Query Language (SQL) ist eine in kommerziellen Systemen weit verbreitete Datenbanksprache, deren Entwicklung das Ziel verfolgte, möglichst einfach für Endbenutzer einsetzbar zu sein. Die Grundstruktur einer Anfrage in \uparrow SQL ist wie folgt:

```
SELECT Attribut(e)
FROM Relation(en)
WHERE Bedingung;
```

Dies ist der umgangssprachlichen Formulierung "Suche Attributwerte in Relation(en), für die Bedingung gilt" nachempfunden. Die Beispielanfrage lautet in SQL:

```
SELECT NAME, REL-ARBZEIT
FROM angestellte, projekte
WHERE angestellte.PROJ-NR=
projekte.PROJ-NR AND angestellte.WOHNORT = "Karlsruhe";
```

- Bei graphikorientierten Sprachen werden Anfragen - nicht wie bei \uparrow Programmiersprachen üblich - in einer linearen Notation formuliert, sondern dadurch, daß auf einem graphischen Bildschirm an entsprechenden Stellen Eintragungen gemacht werden. In Query by Example (QBE) benennen Benutzer zunächst die Relation oder Relationen, mit welchen gearbeitet werden soll. Das System stellt auf dem Bildschirm eine leere Tabelle mit allen Attributbezeichnungen zur Verfügung, etwa nach dem Beispiel der folgenden Tabelle.

angestellte	ANG-NR	NAME	WOHNORT	ABT-NR

Die Anfrage wird nun dadurch spezifiziert, daß ein Beispieltupel für eine mögliche Antwort in diese Tabelle (bzw. Tabellen) eingesetzt wird. Ein Beispieltupel besteht aus Konstanten und Variablen, wobei die Variablen als Beispieltwerte aus der entsprechenden Spalte entsprechenden Domain angegeben werden. Zur optischen Unterscheidung der Konstanten von den Variablen werden die Variablen unterstrichen. Einträge in die Bei-

spieltabelle können mit "P." gekennzeichnet werden. Dadurch wird ausgedrückt, daß die entsprechenden Daten auf dem Bildschirm ausgegeben werden. Stammen die auszugebenden Werte aus mehreren Relationen, so muß eine geeignete Ergebnistabelle generiert werden. Die Anfrage kann in QBE nach dem Beispiel formuliert werden.

angestellte	ANG-NR	NAME	WOHNORT	ABT-NR
	300	MEYER	KA	
ang-pro1	P-Nr	ANG-NR	PROZ-ARBZEIT	
	770008	300	10	
ergebnis	NAME	WOHNORT	PROZ-ARBZEIT	
	P.MEYER	KA	P.10	

Literatur: Atzeni, P., DeAntonellis, V.: Relational Database Theory, Redwood City, 1993. Codd, E. F.: The Relational Model for Database Management. Version 2, Reading/Massachusetts, 1990. Mayer, D.: The Theory of Relational Databases, Rockville, 1983. Paradaens, J. et al.: The Structure of the Relational Database Model, Berlin, 1989. Zloof, M. M.: Query-By-Example: The invocation and definition of tables and forms, in: Kerr, D.S. (Ed.): Proc. of the Int. Conference on Very Large Data Bases, Framingham/Massachusetts, 1975.
Prof. Dr. W. Stucky, Dr. A. Oberweis, Karlsr.

Ablauf- und Terminplanung

Ablauforganisation, mit der Arbeitsinhalte des Vorhabens (Was ist zu tun?), Arbeitszeiten (Wann ist etwas zu tun?) und Arbeitszuordnungen (Mit wem ist etwas zu tun?) geplant werden.

Abstraktion

Vernachlässigung von Details und Konzentration auf wesentliche Aspekte; modellhafte Darstellung der Realität, wobei durch Vernachlässigung der Details die Komplexität vermindert wird.

ACM

Association for Computing Machinery

Administrationsintegration

Zwischenbetriebliche Verketzung von Administrationssystemen in der Wertschöpfungskette.

Administrationssysteme

Systeme zur rationellen Verarbeitung von Massendaten der Verwaltung.

Aggregation

Gerichtete Beziehungsmenge zwischen Klassen; Assoziation mit spezieller Semantik; Objekte, die einzelne Komponenten repräsentieren, werden mit dem Objekt assoziiert, das das gesamte Aggregat darstellt.

Aiken, Howard H.

Entwicklung der Rechenmaschine Harvard Mark I 1944 mit dekadischen Zählrädern (15 m lang, 2,5 m hoch, 80 km Leitungsdraht, 700.000 Einzelteile). Die Multiplikation zweier 10stelliger Zahlen dauerte 6 Sekunden.

Akustikkoppler

Gerät zur Datenübertragung über das öffentliche Telefonnetz.

Akzeptanz

Verhaltensweise des Nutzers (Benutzers, Anwenders) gegenüber dem System. Sie wird nach Einstellungs- und Verhaltensakzeptanz der Technikverwendung bzw. nach Hardware- und Softwareergonomie unterschieden.

Algol

Algorithmic Language; imperative Programmiersprache für den technisch-wissenschaftlichen Anwendungsbereich.

Analogrechner

Anlage, die zur Darstellung von Daten physikalische Größen verwendet.

Analysemethoden, objektorientierte

Begriffserklärung

Eine Analysemethode ist eine zielgerichtete Vorgehensweise, um die Anforderungen an ein Softwaresystem zu ermitteln und zu spezifizieren. Seit 1990 wurden mehrere Methoden der objektorientierten Analyse publiziert. Sie bauen auf den Konzepten der objektorientierten Programmiersprachen (Objekt, Klasse, Attribut, Operation, Vererbung, Botschaft), der Datenmodellierung (Assoziation, Aggregation) und des Zustandsautomaten (Objektlebenszyklus) auf.

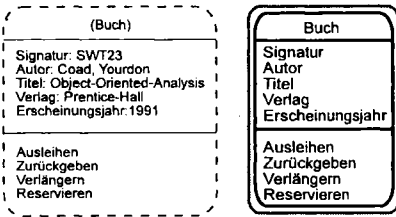
Zielsetzung

Ein großer Vorteil der objektorientierten Systemanalyse liegt darin, daß die meisten Konzepte durchgängig in den nachfolgenden Phasen des Entwurfs und der Implementierung verwendet werden. Dadurch werden Strukturbrüche zwischen

den Phasen vermieden. Die objektorientierte Analyse unterstützt besonders die realitätsnahe \uparrow Modellierung von Fachkonzepten, die Änderbarkeit und Erweiterbarkeit eines realisierten \uparrow Systems und die Wiederverwendbarkeit existierender \uparrow Software.

Inhalt

- Ein Objekt ist ein individuelles, eindeutig identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt.

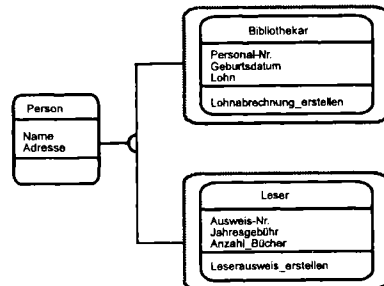


Jedes Buch besitzt bestimmte Eigenschaften (Attributwerte), z.B. Titel = Object Oriented Analysis. Auf das Buchobjekt sind die Funktionen "Ausleihen", "Zurückgeben", "Verlängern" und "Reservieren" anwendbar. In der objektorientierten Terminologie werden sie als Operationen oder Methoden bezeichnet. Die Attributwerte des Objekts können nur mittels seiner Operationen gelesen und geändert werden. Das bedeutet, daß die interne Repräsentation der Objekteigenschaften nach außen verborgen ist. Objekte realisieren das Konzept der Datenkapselung. Das eindeutige Identifizieren der Objekte erfolgt mittels des \uparrow Attributs Objekt-ID, das jedoch nicht in das Diagramm eingetragen wird.

- Alle Objekte mit den gleichen Eigenschaften und gleichen Operationen gehören zu derselben Klasse. Außerdem muß festgelegt werden, wie ein neues Objekt dieser Klasse erzeugt wird. Die gerastete Linie gibt wiederum an, daß diese Klasse Objekte besitzt.
- \uparrow Attribute beschreiben die Eigenschaften einer \uparrow Klasse. Alle \uparrow Objekte einer Klasse besitzen dieselben Attribute, sie unterscheiden sich jedoch in der Regel durch ihre Attributwerte.
- \uparrow Operationen beschreiben das Verhalten der Objekte. Eine Operation ist ei-

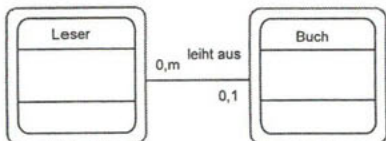
ne ausführbare Tätigkeit im Sinne einer Funktion bzw. eines Algorithmus und kommuniziert mit der Umgebung über Ein-/Ausgabeparameter.

Die \uparrow Vererbung ist eine gerichtete Beziehung zwischen \uparrow Klassen, wobei eine Unterklasse alle \uparrow Attribute und \uparrow Operationen ihrer Oberklasse übernimmt. Es entsteht eine Klassenhierarchie bzw. eine Vererbungsstruktur. Die Klasse "Leser" erbt alle Attribute der Klasse "Person". Somit kann jedes Objekt der Klasse "Leser" Attributwerte für "Name", "Adresse", "Ausweis-Nr", "Jahresgebühr" und "Anzahl Bücher" besitzen. Analog dazu erbt die Klasse "Bibliothekar" die Attribute der Klasse "Person". Operationen können genau wie Attribute vererbt werden. Die Klasse "Person" wird als Oberklasse der Klassen "Leser" und "Bibliothekar" bezeichnet. Entsprechend heißen "Leser" und "Bibliothekar" Unterklassen von "Person". Es fehlt der Klasse "Person" die gerastete Linie. Das bedeutet, daß für diese Klasse keine Objekte erzeugt werden können. Derartige Klassen heißen abstrakte Klassen. Das Verwenden der abstrakten Klasse bedeutet, daß es - bei der beschriebenen Modellbildung - keine Person gibt, die weder Leser noch Bibliothekar ist. Es wird zwischen den Konzepten der Einfachvererbung und der Mehrfachvererbung unterschieden. Liegt eine Einfachvererbung vor, dann besitzt jede Klasse max. eine direkte Oberklasse. Es entsteht eine Baumstruktur. Bei der Mehrfachvererbung kann jede Klasse mehrere direkte Oberklassen besitzen. Es entsteht eine Netzstruktur.

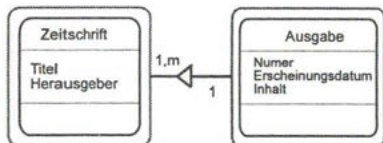


Eine \uparrow Assoziation ist eine Beziehungsmenge zwischen Klassen. Sie ist inhärent bidirektional und modelliert die Menge

der Beziehungen zwischen einzelnen Objekten der beteiligten Klassen. Die Assoziation "leiht aus" modelliert eine Beziehung zwischen einem "Leser" und einem "Buch". Die Kardinalität bezeichnet die Wertigkeit einer Beziehung zwischen Klassen. Sie sagt aus, wieviele Objekte an der Beziehung beteiligt sind. Bspw. kann ein Leser kein Buch ausleihen oder auch mehrere Bücher ausleihen. Dies wird durch die Kardinalität "0,m" angegeben. Jedes Buch kann maximal von einem Leser ausgeliehen werden. Dementsprechend wird die Kardinalität mit "0 oder 1" spezifiziert. Kardinalitäten können numerisch oder graphisch angegeben werden. Es existieren zahlreiche Notationen, die sich meistens an Entity-Relationship-Notationen orientieren.

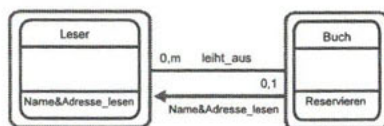


Eine \uparrow Aggregation ist eine gerichtete Beziehungsmenge zwischen Klassen, d.h. eine Assoziation mit spezieller Semantik. Objekte, die einzelne Komponenten repräsentieren, werden mit dem Objekt assoziiert, das das gesamte Aggregat darstellt. Bspw. gehören zu einer Zeitschrift (Aggregat) eine oder mehrere Ausgaben (Komponenten). Die Kardinalität wird mit "1 oder m" angegeben. Jede Ausgabe gehört zu genau einer Zeitschrift. Die Kardinalität ist "1". Der Titel und der Herausgeber sind Eigenschaften der Zeitschrift; die Nummer, das Erscheinungsdatum und der Inhalt sind Eigenschaften einer einzelnen Ausgabe sind.

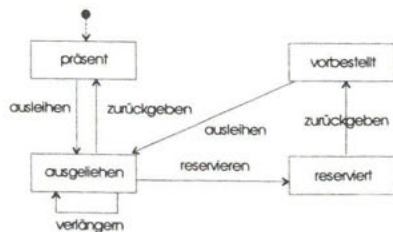


Eine \uparrow Botschaft ist die Aufforderung eines Senders an den Empfänger, eine Dienstleistung zu erbringen. Sender und Empfänger sind i.a. Objekte, können aber auch Klassen sein. In der Bibliotheksver-

waltung können ausgeliehene Bücher reserviert werden. In diesem Fall wird der Leser aufgefordert, das Buch zurückzugeben. Bei Abarbeitung der Operation "Reservieren" schickt das Buchobjekt dem zugehörigen Leserobjekt die Botschaft "Name&Adresse_leesen".



Der \uparrow Objektlebenszyklus modelliert für ein Objekt die möglichen Abarbeitungsreihenfolgen der Operationen. Alle Objekte einer Klasse verhalten sich gleich. Dementsprechend ist es ausreichend, pro Klasse einen Objektlebenszyklus zu erstellen. Jedes Objekt ist in einem individuellen Zustand und wechselt unabhängig von anderen Objekten von einem Zustand zum nächsten. Ein Buch ist entweder "präsent", "ausgeliehen", "reserviert" oder "vorbestellt". In Abhängigkeit vom aktuellen Zustand des Buches sind nur bestimmte Operationen ausführbar. So kann ein Buch nur dann reserviert werden, wenn es ausgeliehen ist.



Hinweis: Für das obige Beispiel wurde die Notation von (Coads, Yourdon, 1991) gewählt. Für das Objektsymbol verwenden die Autoren eine eigene Erweiterung dieser Notation.

Methoden

Viele Methoden werden durch \uparrow CASE-Werkzeuge unterstützt. Die einzelnen Methoden besitzen unterschiedliche Schwerpunkte. Im konkreten Anwendungsfall kann eine Methodensynthese - wie in (Balzert, 1994; Stein, 1994) beschrieben - zweckmäßig sein.

Makro-komponenten	Methodenkomponenten	Wirfs-Brock	Booch	Coad	Rumbaugh	Shlaer	Embley	Rubin	Nelson	Jacobson	Martin	Henderson	Sully	Dillon
		Methode												
Basis-komponenten	Konkrete Objekte		X		X									
	Attribute		X	X	X	X			X	X	X	X		X
	Operationen	X	X	X	X	X		X	X		X	X	X	X
	Klassen	X	X	X	X	X	X		X	X	X	X	X	X
Struktur-komponenten	Vererbung	X	X	X	X			X	X	X	X	X	X	X
	Assoziation		X	X	X		X		X	X	X	X	X	X
	Aggregation	X	X	X	X		X		X	X	X	X	X	X
	Botschaftsaustausch	X	X	X			X	X	X	X	X	X	X	X
Dynamik-komponenten	Subsysteme	X	X	X	X			X	X	X	X	X		X
	Botschaften	X	X	X	X		X	X	X					X
	Objektlebenszyklen		X		X		X			X	X		X	X
	Dynamisches Systemverhalten		X		X			X	X					
Echtzeit-komponenten	Zeitanforderungen		X						X					X
	Schnelle Reaktionen				X		X							
	Prioritäten						X							

Legende: = Konzept wird unterstützt

Literatur: Balzert, H.: Methoden der objektorientierten Systemanalyse, Mannheim, 1994. Balzert, H., Stein, W.: Worin unterscheiden sich die objektorientierten Analysemethoden?, in: OBJEKTspektrum, Bd. 1, Heft 2, Mai/Juni 1994. Booch, G.: Object Oriented Analysis & Design with Applications, 2nd. ed., Redwood City, CA, 1994. Coad, P., Yourdon, E.: Object-Oriented Analysis, 2nd. ed., NJ, 1991. Dillon, T., Tan, P. L.: Object-Oriented Conceptual Modeling, Sydney, 1993. Stein, W.: Objektorientierte Analysemethoden - Vergleich, Bewertung, Auswahl, Mannheim, 1994. Sully, P.: Modelling the World with Objects, London, 1993.

Prof. Dr. H. Balzert, Dr.-Ing. W. Stein,
Dortmund

Anfrage, natürlichsprachige

Versuch, den dem Benutzer vertrauten sprachlichen Kommunikationsmodus für die Suchformulierung zu verwenden; werden vor allem im Bereich der Künstlichen Intelligenz entwickelt; Grundgedanke ist, daß sich der Benutzer in der Mensch-Computer-Interaktion analog zur zwischenmenschlichen Kommunikation verhält.

Angewandte Informatik

†Informatik, Angewandte

Animationen

Sequenz von abgeänderten Zeichnungen/ Grafiken/Fotos und/oder Texten; werden innerhalb kurzer Zeit überblendet; auf

diese Weise wird für das menschliche Auge ein Bewegungseffekt simuliert; Bildfrequenz ist sehr von den Anforderungen abhängig, liegt aber meist in dem Bereich von 16 - 25 Bilder/Sek.

Animationsprogramm

Programm zur Erzeugung von Zeichnungen z.B. auf Videofilmen, die in Graphikformat gespeichert werden.

ANSI

American National Standards Institute

ANSI/SPARC

Modell; beinhaltet eine Empfehlung, welche Bausteine ein †DBS bzgl. ihrer nach außen sichtbaren Funktionen haben soll und wie die Bausteine über verschiedene †Schnittstellen zusammenspielen.

Anwendung, verteilte

Menge von Softwarekomponenten, die räumlich verteilt auf verschiedenen Knoten eines †Rechnernetzes ablaufen und koordiniert kooperieren.

Anwendungssoftware

Technisch-wissenschaftliche, kommerzielle und prozeßsteuernde, auf einzelne Funktionen bezogene Programme.

Anwendungssoftware für das Rechnungswesen

Teilgebiete

Die A. für das Rechnungswesen wird üblicherweise in zwei Bereiche unterteilt, die ihrerseits wieder in mehrere Unterbereiche zerfallen. Der eine Teil befaßt sich

mit der ↑Information Unternehmensexterner, wie z.B. Gesellschafter, Gläubiger, Finanzbehörden (externes Rechnungswesen); der andere mit der Information des ↑Managements eines Unternehmens (internes Rechnungswesen).

Das externe Rechnungswesen wird weiter in die Erstellung des Hauptbuches (Bilanz, Gewinn- und Verlustrechnung) und in die Erstellung verschiedener Nebenbücher unterteilt. Ein Nebenbuch erklärt jeweils ein Konto des Hauptbuches. Beispiele für Nebenbuchhaltungen sind die Debitoren- und die Kreditorenbuchhaltung, die Anlagen- und die Materialbuchhaltung. Ein weiteres Teilgebiet des externen Rechnungswesens ist die Konsolidierung. Hier werden die Abschlüsse der einzelnen Gesellschaften zum Konzernabschluß zusammengeführt. Datnmäßige Grundlage für das externe Rechnungswesen sind Vorgänge in der Leistungssphäre des Unternehmens (Materialkäufe, Produktverkäufe, ...), die bewertet werden, Vorgänge in der Finanzsphäre (Ein- und Auszahlungen).

Auch das interne Rechnungswesen wird in Teilgebiete untergliedert. Diese orientieren sich im allgemeinen an der Kostenüberwälzung:

- Das Gemeinkostencontrolling befaßt sich mit der Planung und Kontrolle der Betriebsbereitschaft. Diesbezügliche Ordnungsbegriffe sind die Kostenstelle, der innerbetriebliche Auftrag (das Projekt) und der Prozeß. Wesentliche Funktionen sind die Gemeinkostenumlage und die innerbetriebliche Leistungsverrechnung.
- Das Produkt- und Produktionscontrolling beschreibt die Leistungserstellung. Fertigungsauftrag, Produkt und Kostenträger sind die entsprechenden Ordnungsbegriffe. Im Vordergrund stehen die Vor- und Nachkalkulationen einer Erzeugniseinheit.
- Mit Hilfe des ↑Vertriebscontrolling wird gesteuert, daß die Leistungen bestmöglich vermarktet werden. Kundenaufträge, Kundenprojekte und beliebig definierbare Marktsegmente sind die entsprechenden Beschreibungsobjekte. Die Steuerung erfolgt, indem für die genannten Beschreibungsobjekte mehrstufig Ergebnisse bzw. Deckungsbeiträge ermittelt werden.

Neben dieser am Abrechnungsgang orientierten Gliederung in Teilmodule steht häufig noch das Bereichscontrolling. Hierbei wird das gesamte Unternehmen in selbständig agierende Einheiten (Profit Centers) unterteilt, in deren Verantwortung alle Funktionen der Betriebsbereitschaft und der Leistungserstellung liegen. Profit Centers werden häufig nach produktorientierten oder regionalen Gesichtspunkten gebildet. Diese Einteilung ist unabhängig von der rechtlichen Gliederung und gleichberechtigt zu dieser. Um für alle diese Ordnungsbegriffe wertmäßige ↑Informationen zur Verfügung zu haben, müssen die zwischen dem Unternehmen und seiner Umwelt auftretenden Geschäftsvorfälle diesen zugeordnet (kontiert) werden, ebenfalls die innerbetrieblichen Lieferungen und Leistungen. Bei diesen muß zusätzlich fiktiv bewertet werden, da originär nur Zeiten und Mengen erfaßt werden. Für diesen Zweck wurden verschiedene Modelle entwickelt (Voll-/ Teilkosten-, Ist-/Standardkosten-, relative Einzelkosten- und Deckungsbeitragsrechnung).

Alle Teilgebiete sowohl des internen als auch des externen Rechnungswesens beinhalten Funktionen zur Beschreibung von Ist-Vorgängen, zur Planung und zum Berichtswesen. Die Planungsaktivitäten sind im internen Rechnungswesen in der Regel umfangreicher, da sie nicht nur der Vorausschau dienen, sondern auch die Voraussetzung für die Verrechnungen zwischen den Ordnungsbegriffen und für die Bestandsbewertung bilden. Die genannten Teilbereiche dienen der softwaretechnischen Modularisierung. Häufig werden auch die Abteilungen entsprechend gegliedert. Inhaltlich bestehen zahlreiche Verbindungen, so bspw. zwischen der Herstellkostenkalkulation und der Bestandsbewertung, bzw. zwischen der Ergebnisrechnung und der Erzeugniskalkulation.

Architektur der Anwendung

Die Systeme des Rechnungswesens sind Teil einer Anwendungsarchitektur, die aus mehreren Ebenen besteht. Auf der untersten Ebene sind die technischen Abwicklungssysteme angesiedelt (z.B. ↑CAD, ↑Betriebsdatenerfassung, Lagerverwaltung, Kassenhaltung). Die Systeme

der Materialwirtschaft (Einkauf, Produktion, Vertrieb), des Personalwesens und der Finanzwirtschaft bilden die nächste Ebene. Auf diesen basieren die Systeme des Rechnungswesens mit den o.g. Teilkomponenten. Die oberste Ebene bilden unternehmensweite ↑Führungsinformationssysteme.

Konzeptionell arbeiten alle diese Systeme nach einem Modell von Ereignis und Bestand:

- Ein Ereignis kann in allen vier Ebenen jeweils mit anderen Beschreibungsmerkmalen enthalten sein, oder es fließt in periodenbezogene Verdichtungen ein.
- Ein Ergebnis hat eine positive und eine negative Bestandswirkung.
- Ein Ergebnis kann mit anderen Ereignissen gekoppelt sein.

Die Rückmeldung einer Tätigkeit für einen Fertigungsauftrag beispielsweise wird im Betriebsdatenerfassungssystem aufgezeichnet; sie bewirkt die Freigabe der Kapazität; sie hat gegebenenfalls Auswirkungen auf die Entlohnungsgrundlage des Mitarbeiters; sie führt zu einer wertmäßigen Entlastung der leistenden Kostenstelle und einer Belastung des Fertigungsauftrags; sie veranlaßt bei einem besonders wichtigen Fertigungsauftrag einen Eintrag im ↑Führungsinformationssystem.

Auch die Systematik der Planungsfunktionen orientiert sich an einem Modell mit mehreren Ebenen. In der Materialwirtschaft werden die logistischen Abläufe geplant (Absatzplan, Planproduktionsprogramm, Beschaffungsplan). Aus diesen leiten sich die Pläne des Rechnungswesens ab. Entsprechend den Abbildungsgrößen wird dort zwischen der Kosten- und Erlösplanung (Planung der Kostenstellen, Erzeugniskalkulation, Kostenträger- und Ergebnisplanung), der Aufwands- und Ertragsplanung (Plan-GuV, Planbilanz) und der Einzahlungs- und Auszahlungsplanung (Finanzplan) unterschieden. Die Ebene der ↑Führungsinformationssysteme enthält die Unternehmensplanung.

Die einzelnen Pläne können isoliert oder durch Referenz zueinander erstellt werden. Innerhalb der einzelnen Pläne können Verdichtungen (Bottom up-Planung) und Verteilungen (Top down-Planung)

vorgenommen werden. Die Planung kann sich jeweils auf kürzere und längere Zeiträume beziehen.

Die Gesamtheit der Einzelereignisse oder deren Periodenverdichtungen - insb. die Plandaten stehen i. d. R. nur als Periodenbeträge zur Verfügung - sowie die korrespondierenden Bestände werden als Grundrechnung bezeichnet. Auf dieser basieren Auswertungsrechnungen. Da insb. im innerbetrieblichen Rechnungswesen die Fragestellungen im Zeitpunkt der Gestaltung der Grunddatensammlung nicht bekannt sind, muß diese zweck- und methodenneutral aufgebaut sein.

Softwaretechnische Aspekte

Im folgenden werden einige ausgewählte Aspekte der softwaretechnischen Lösung beschrieben:

- Es erweist sich als sehr vorteilhaft, wenn die einzelnen Ebenen der Anwendungsarchitektur stets in einem zueinander konsistenten Zustand sind. Die Bezugnahme auf einzelne Ereignisse ermöglicht eine Fortschreibung im Real Time Mode, d.h. ein in der Realität auftretendes Ereignis wird einmal erfaßt und zeitgleich mit seinen Auswirkungen in den jeweiligen Ebenen fortgeschrieben.

- Die Software sollte datenbankorientiert sein, da hierbei die größte Chance besteht, für neu auftretende Fragestellungen entsprechende Unterstützung durch die Daten des Rechnungswesens zu erhalten. Insb. ↑relationale Datenbanksysteme verfügen wegen der universellen Auswertbarkeit ihrer Tabellen und wegen ihrer Datenunabhängigkeit über mächtige Abfragesprachen.

- Die ↑Software zur Unterstützung der Funktionen des Rechnungswesens wird in fast allen Unternehmen nicht erstellt, sondern als ↑Standardsoftware vom Markt bezogen. Diese Standardsoftware muß durch ↑Parametrisierung, d.h. ohne in die Programme eingreifen zu müssen, den individuellen betrieblichen Gegebenheiten anpaßbar sein (↑Customizing). Diese können bspw. aus den gesetzgeberischen Vorschriften des Landes, aus den Gepflogenheiten der Branche oder des Unternehmens (hierzu gehören auch die unterschiedlichen Systeme der Kostenrechnung) oder aus den Abläufen am einzelnen Arbeitsplatz resultieren.

- Voraussetzung hierfür ist, daß das gesamte Anwendungssystem modular aufgebaut ist mit offen verfügbaren ↑Schnittstellen. Dieses Architekturmerkmal ermöglicht den gleichzeitigen Einsatz von ↑Anwendungssoftware verschiedener Hersteller. Außerdem gestattet es eine Verteilung auf mehrere Rechner bei gleichzeitiger betriebswirtschaftlicher Integration. Diesbezügliche Verteilungsszenarien sind bspw. dezentrale Logistik bei zentralem Rechnungswesen oder dezentrales ↑Vertriebscontrolling bei zentraler Ergebnisrechnung.
 - Die Standardsoftware muß hardware-unabhängig sein. Hier hilft eine softwaretechnische Trennung von hardwareabhängigen und -unabhängigen Teilen.
 - Aus der hardwaretechnischen Spezialisierung resultiert eine softwaretechnische Trennung dahingehend, daß bestimmte Funktionen auf Maschinen gelegt werden können, von diesen besonders effizient unterstützt werden. Im Sinne eines ↑Client/ Server-Systems hat sich hier eine Aufteilung der Software in Datenbank-, Verarbeitungs- und Präsentationsfunktionen als zweckmäßig erwiesen.
- Literatur: Scheer, A.-W.: Wirtschaftsinformatik - Referenzmodelle für individuelle Geschäftsprozesse. 4. Aufl., Berlin, 1994. Sinzig, W.: Datenbankorientiertes Rechnungswesen - Grundzüge einer EDV-gestützten Realisierung der Einzelkosten- und Deckungsbeitragsrechnung. 3. Aufl., Berlin, 1990. Wedekind, H.: Kaufmännische Datenbanken, Mannheim, Leipzig, 1993.*

Dr. W. Sinzig, Walldorf

Anwendungssysteme

A. stellen Softwaresysteme dar, mit deren Hilfe die computergestützte ↑Informationsverarbeitung in den verschiedenen Anwendungsbereichen zur Durchführung kommt. Die *A.* bzw. die Anwendungsprogramme dienen somit der Eingabe und der Transformation der ↑Informationen, der Aufbereitung und der Ausgabe der Ergebnisse. Weiterhin lassen sich mit Hilfe von *A.*-↑Informationen ablegen (speichern), wiederfinden und übertragen. Die Basis der *A.* bilden Programme, die entsprechend ihrer Zielsetzung mit Hilfe unterschiedlicher Sprachen entwickelt wurden.

Neben der *A.* (Application Programs; User Programs), wird die ↑Software auch in die Systemsoftware bzw. Systemprogramme eingeteilt (System Programs). Die Kernkomponente der Systemsoftware stellt das ↑Betriebssystem (Operating System) dar, das die Abwicklung der *A.* steuert und kontrolliert, wobei die effiziente, effektive Nutzung der DV-Ressourcen zu gewährleisten ist (Hansen, 1992; Mertens, Bodendorf u. a., 1991; Scheer, 1990; Stahlknecht, 1993; Zilahi-Szabó, 1993).

A. stellen somit die Softwaresysteme dar, die die Ausführung unterschiedlicher Aufgaben der ↑Informationsverarbeitung bzw. die Herleitung von Problemlösungen direkt unterstützen. Der Einsatz von *A.* stellt deshalb den eigentlichen Zweck der Informationsverarbeitung dar bzw. soll unmittelbar zu einem Nutzen führen. Da *A.* die Leistungspotentiale der automatisierten ↑Datenverarbeitung, wie z.B. hohe Verarbeitungsgeschwindigkeit, große Speicherkapazitäten mit direktem Zugriff und genaues und unermüdliches Verarbeiten nutzen, führt der Einsatz dieser Programme zu großen Vorteilen, auf die heute in der Wirtschaft und Verwaltung, sowie in Wissenschaft und Forschung nicht mehr verzichten werden kann. Neben dem Einsatz in operativen Anwendungsbereichen gewinnen die *A.* auch eine ständig wachsende, größere strategische Bedeutung.

Bevor *A.* genutzt werden kann, müssen sie konzipiert und in einer ↑Programmiersprache formuliert werden. Mit dem Aufbau und der Entwicklung von *A.* beschäftigt sich das ↑Software Engineering, das darüber hinaus Wartung und Pflege, sowie die Weiterentwicklung der Programmsysteme betrachtet. Das Software Engineering behandelt somit den gesamten Lebenszyklus eines *A.* Es stützt sich dabei auf Prinzipien, ↑Methoden, Werkzeuge (↑Tools) und Programmiersprachen, die sich durch ein ↑CASE-System (Computer Aided Software Engineering) organisieren und einsetzen lassen.

Der Einsatz von *A.* orientiert sich an einem Bedarf. Nach einer eingehenden Problemanalyse und Anforderungsdefinition (bzw. nach einer Ist-Analyse und Sollkonzeption) stellt sich die grund-

gende Frage, ob das *A.* eigenentwickelt oder fremdbezogen werden soll. Durch eine individuelle Programmierung ist einerseits eine sehr gute Anpassung an die jeweilige Aufgaben- bzw. Problemstellung möglich, andererseits setzt die Eigenentwicklung jedoch auch das Vorhandensein der notwendigen Ressourcen, wie z.B. Maschinen, Sprachen und Programmierer voraus. Der Fremdbezug von Software läßt sich sowohl durch Kauf von ↑Standardsoftware als auch durch eine Auftragsprogrammierung realisieren, die von einem Software- bzw. Systemhaus (so z.B. auf der Basis eines ↑Outsourcing-Konzeptes) durchgeführt werden kann. Beim Aufbau von größeren *A.* in der Praxis, der in Form von Projekten organisiert wird, finden sich bei einer Systementwicklung i.d.R. sowohl Eigenentwicklung (↑Individualsoftware) als auch Fremdbezug von Teilsystemen in Form des Kaufs von ↑Standardsoftware und individueller Auftragsfertigung.

Der Aufbau von *A.* ist häufig mit hohen Kosten verbunden, die im Rahmen von Beschaffung bzw. Entwicklung und insb. auch von Pflege und Wartung der Software entstehen. Der Einsatz der Systeme soll Nutzen erbringen, der sich teilweise erst langfristig einstellt oder in Form von Wettbewerbsvorteilen auftreten kann. Die Entscheidung für die Anschaffung bzw. Entwicklung eines *A.* stellt ein Investitions- und Finanzierungsproblem dar, das eine gründliche Wirtschaftlichkeitsanalyse voraussetzt, eine systematische Gegenüberstellung von Kosten und Nutzen.

A. lassen sich nach verschiedenen Kriterien einteilen, so z.B.:

- nach den notwendigen DV-Systemen, d.h. ↑Hardware- und ↑Betriebssystemen, Hardware-Plattformen wie Groß-, bzw. Abteilungsrechner-, ↑Workstations- und ↑Personal Computerplattformen und ↑Betriebssystemplattformen, so z.B. die Großrechnersysteme VM, MVS oder BS 2000, ↑UNIX, ↑OS/2 bzw. ↑MS-DOS);
- nach Einsatz- bzw. Anwendungsbereichen, wie technische bzw. ingenieurwissenschaftliche, mathematische und naturwissenschaftliche, kaufmännische, verwaltungstechnische bzw. betriebswirtschaftliche Anwendungen (weiterhin sind Anwendungen in der Rechtswissenschaft,

den Sozialwissenschaften und der Medizin gegeben);

- nach betrieblichen Funktionsbereichen, wie z.B. Beschaffung, Logistik, Lagerhaltung, Produktion, Verkauf, Marketing, ↑Controlling, Rechnungswesen, Personalwesen. Diese Einteilung wird in der Praxis häufig genutzt. Verbreitete Systeme sind Materialwirtschafts-, Produktionsplanungs- und Steuerungssysteme (↑PPS-Systeme), Kostenrechnungs-, Vertriebssteuerungs- und ↑Personalinformationssysteme;
- nach Wirtschaftsbranchen und öffentlichen Verwaltungsbereichen, wie z. B. Industrieunternehmen, Handelsunternehmen, Banken, Versicherungen, Handwerksbetriebe, Beratungsunternehmen, öffentliche Verwaltungen (auch diese Einteilung in die sog. ↑Branchensoftware wird in der Praxis häufig genutzt);
- nach Funktion der ↑Informationsverarbeitung, die durch das *A.* hauptsächlich unterstützt wird, so z.B. ↑Anwendungssoftware zur Eingabe der Daten (Erfassungssysteme), zur Aufbereitung und Ausgabe der Ergebnisse (Präsentationssoftware, graphische Systeme), zur Informationstransformation (z.B. algorithmische Programme zur mathematischen bzw. statistischen Berechnung), zur Informationsspeicherung (hauptsächlich in Form von Datenbanksoftware) und zur Übertragung von Informationen (Kommunikationssoftware);
- nach ↑Standardsoftware (Packaged Software) und ↑Individualsoftware (Custom Software);
- nach proprietären Softwaresystemen (herstellerspezifischen Systemen) und offenen Systemen, die nach herstellerunabhängigen Standards aufgebaut sind;
- nach Gestaltung der möglichen ↑Benutzeroberflächen (bei ↑Dialogsystemen), wie z.B. Menüsysteme oder graphische Systemoberflächen;
- nach der eingesetzten ↑Programmiersprache, wie z.B. Assembler, prozedurale Programmiersprache, 4GL-System, objektorientierte, funktionale oder logische Sprache, oder nach den unterschiedlichen Werkzeugen und Entwicklungsumgebungen (↑CASE-Tools);
- nach ↑Betriebsarten des DV-Systems (in Abhängigkeit vom ↑Betriebssystem),

so z.B. *A.* auf der Basis von Einprogramm- oder Mehrprogrammbetrieb, von Einbenutzer- oder Mehrbenutzerbetrieb (Multiuser System) oder auf der Basis von ↑Stapelverarbeitungs- oder interaktiven Verarbeitungs- (↑Dialogsystemen);

- nach Nutzungsmöglichkeiten in Unternehmen in Form von Administrations-Transaktions-, Dispositions-, Planungs- und ↑Kontrollsystemen, so z.B. als Entscheidungs- und ↑Managementunterstützungssysteme;

- nach geschlossenen (auf einem Rechner ablaufenden) und verteilten *A.*, die auf der Basis von ↑Rechnernetzen verteiltes und kooperatives Arbeiten unterstützen, so in Form von ↑Groupware und ↑Computer Supported Cooperative Work (CSCW);

- nach der verwendeten Technologie in *A.*, die sich auf sequentielle Programme oder auf parallele Programme stützen, oder die auf konventionellen oder auf objektorientierten bzw. wissensbasierten Softwaresystemen (z.B. ↑Expertensysteme) basieren.

Ziel bei der Beschaffung von *A.* ist heute immer mehr der Einsatz von standardisierten, offenen und integrierten Softwaresystemen mit klaren, standardisierten ↑Schnittstellen und einheitlichen Kommunikations- und Benutzerunterstützungen. So haben große DV- und Softwarehersteller bzw. -anbieter einheitliche Rahmenwerke für die unternehmensweite Anwendungsintegration geschaffen, die weltweit bekannt sind. Ziel einer solchen Anwendungsarchitektur (Application Architecture) ist es, *A.* in integrierter Form und nach einheitlichen Richtlinien zu entwickeln, so daß sie auf der Basis von Standardprotokollen miteinander kommunizieren können, eine einheitliche ↑Benutzeroberfläche aufweisen (↑Common User Access, CUA) und auf unterschiedlichen Plattformen (Hard- und Systemsoftware) ablaufen können. Erfolgreiche *A.* sind modular aufgebaut, so daß sie auch flexibel beschafft und bedarfsorientiert genutzt werden können.

Literatur: Hansen, H.R.: Wirtschaftsinformatik I, 6. Aufl., Stuttgart, 1992. Kurbel, K., Strunz, H. (Hrsg.): Handbuch Wirtschaftsinformatik, Stuttgart, 1990. Mertens, P. (Hrsg.): Lexikon der Wirt-

schaftsinformatik, 2. Aufl., Berlin, 1990. Mertens, P. et al.: Grundzüge der Wirtschaftsinformatik, Berlin, 1991. Scheer, A.-W.: EDV-orientierte Betriebswirtschaftslehre, 4. Aufl., Berlin, 1990. Stahlknecht, P.: Einführung in die Wirtschaftsinformatik, 6. Aufl., Berlin, 1993. Zilahi-Szabó, M.G.: Wirtschaftsinformatik, München, 1993.

Prof. Dr. R. Gabriel, Bochum

Anwendungssysteme in Dienstleistungsbetrieben

Traditionell wurden unter Dienstleistern hauptsächlich

- Banken, Versicherungen, Handel,
- Beratungsunternehmen, Hotels und Gaststättengewerbe etc.

verstanden. Diese Einteilungsform ist in der Betriebswirtschaftsliteratur auch heute noch vorherrschend. Für Zwecke der ↑Wirtschaftsinformatik, sowohl für die betrieblichen Anwendungsaufgaben, wie für die Forschung, sollte die traditionelle Gliederung der Fülle neuer ↑Dienstleistungen und Unternehmensformen angepaßt und erweitert werden um:

- klein- und mittelständische Unternehmen, deren Produktion ganz oder teilweise aus Dienstleistungen besteht;

- öffentliche Dienstleister, die sich z.T. auf dem Weg in die Privatisierung befinden, wie die gelbe Post, Telekommunikationsunternehmen, Bahn u.a. Unternehmen, Krankenhäuser, Versorgungsbetriebe und dabei den Übergang in ein betriebswirtschaftliches Management zu leisten haben;

- der Öffentliche Dienst, also alle Behörden der verschiedenen Verwaltungsebenen mit ihren nachgeordneten Ämtern und Anstalten, die seit den 80er Jahren den Weg vom Verwaltungsvollzug in Richtung auf ein Dienstleistungsverständnis zu nehmen beginnen und bei denen inzwischen betriebswirtschaftlich orientiertes Management Einzug zu halten beginnt.

Die Wirtschaftsinformatik der letztgenannten drei Gruppen

- klein- und mittelständische Unternehmen mit Dienstleistungscharakter,
- Öffentliche Dienstleister und
- Öffentlicher Dienst

unterscheidet sich wesentlich von den in den klassischen Bereichen von Banken,

Versicherung und Handel herrschenden Verhältnissen und verdient daher selbständige Betrachtung. Der gesamte genannte Dienstleistungsbereich, außer dem traditionellen Bereich Banken, Versicherungen und Handel, ist bzgl. wirtschaftsinformatischer Fragen in Literatur und Forschung im Vergleich zum zahlenmäßigen Anteil an der Gesamtwirtschaft noch stark unterrepräsentiert. Es gibt zwar eine Fülle von Anwendungssoftwarelösungen auf diesem Markt, aber die wissenschaftliche Aufarbeitung und anwendungsbezogene Unterstützung durch die Wirtschaftsinformatik steht - anders als im Bereich von Produktion und Industrie und klassischer Dienstleistung - noch weitgehend aus.

Klein- und mittelständische Unternehmen als Dienstleister

Historisch wurden in diese Gruppe vor allem Hotels, Gaststätten, Steuerberatungs- und andere Beratungsunternehmen, Reisebüros etc. gezählt. Dieser Bereich hat in den letzten 20 Jahren zwei wesentliche Entwicklungen erfahren, die das gesamte Bild veränderten:

- Seit den 80er Jahren hat sich die Zahl der kleinen Dienstleistungsanbieter und das Spektrum der Dienstleistungsprodukte vervielfacht. Dabei sind viele Anbieter hinzugekommen, die traditionell dem Handwerk zuzurechnen sind, die aber immer stärkere Dienstleistungskomponenten anbieten, die inzwischen öfters schon überwiegen.

- Erst mit dem Auftreten des PC hält die Datenverarbeitung ihren Einzug auch in den Bereich der angesprochenen Unternehmen mit weniger als 100 Mitarbeitern. Dieses Segment war bis Anfang der 80er Jahre für die Datenverarbeitung so gut wie bedeutungslos und ist es für die traditionelle "große EDV" bis heute.

Vor allem solche Unternehmen, die Dienstleistung und Handel, Dienstleistung und Handwerk bzw. kleinindustrielle Fertigung oder gar alle drei Bereiche verknüpfen, sind ein weitgehend neues Phänomen. Typisch sind z.B. kleine Anlagenbauer mit überwiegendem Kundendienstgeschäft oder Baurenovierer, die alte Türen und Fenster nach Katalog erneuern und dabei nur noch eine minimale fertigende Komponente zeigen.

Hier fällt es immer schwerer, diese Unternehmen einer der klassischen Kategorien zuzuordnen. Für die *betriebliche Informationsverarbeitung* bedeutet dies oft, daß nebeneinander Aufgaben des/der

- Handels mit Einkauf und Vertrieb,
- Fertigung, Lagerhaltung, Lieferung,
- des Kundendienstes

zu bearbeiten sind - und dies oft in Betrieben mit weniger als 10 Mitarbeitern. Der Einzug und die Nutzung der Informationsverarbeitung in Unternehmen unter 100 Mitarbeitern unterliegen völlig anderen Bedingungen und Regeln als in den größeren Unternehmen, die traditionell mit Datenverarbeitung ausgestattet sind. Aus Kostengründen ist davon auszugehen, daß Betriebe unter 70 Mitarbeitern im Regelfall nicht in der Lage sind, einen hauptberuflichen EDV-Mitarbeiter zu bezahlen. Dies bedeutet, daß die in größeren Betrieben für Stabilität der DV-Leistungen sorgende Organisationseinheit ↑Rechenzentrum nicht zur Verfügung steht. Damit sind alle traditionellen Organisationsformen der Datenverarbeitung nicht in die Kategorie der betrachteten kleinen Unternehmen übertragbar. Alles, wofür in der großen EDV in den Bereichen Produktion und Handel ausgereifte Lösungen vorhanden sind, muß völlig neu bedacht, bearbeitet und organisiert werden.

Sondersituation

Die Situation dieser Gruppe wird entsprechend durch zwei Komponenten charakterisiert:

- durch die allgemeine Problematik der Klein- und Mittelbetriebe im Umgang mit ↑Informationsverarbeitung;
- durch die Vielfalt der miteinander logisch vernetzten betrieblichen Funktionen, wenn Komponenten aus Produktion, Handel und anderen Dienstleistungen zusammentreffen.

Bei dezentralisierten Konzerntöchtern, die immer häufiger in die ausgesprochene Größenordnung eindringen, ist die angesprochene Problematik der Informationsverarbeitung meist nicht vorhanden, da zentrale Ressourcen des Unternehmens Hilfestellung geben.

Die klassischen Engpässe

Die Gruppe der dienstleistenden Klein- und Mittelbetriebe zeigt weitgehend die-

selben Risikofelder: defizitäre kaufmännische, technische und betriebsorganisatorische Vorbildung der Betriebsleitung, die sich äußert durch

- mangelnde Fähigkeit, betriebswirtschaftlich anspruchsvolle Software zu nutzen;
- mangelnde Vorbildung für Hardware- und Softwareentscheidungen;
- mangelnde Einsicht in die Ausbildungserfordernisse für Mitarbeiter;
- mangelnde Fähigkeit zur organisatorischen Umgestaltung bei DV-Einsatz;
- eine hohe Beratungsbedürftigkeit, die auf zwei Engpässe führt, auf die mangelnde Erfahrung des Umgangs mit Beratern und auf die oft gezeigte Hilflosigkeit im Umgang mit Anbietern von Hard- und Software (die noch dadurch verstärkt wird, daß viele Betriebsberater von den Anbietern Provisionen erhalten);
- den Mangel an der persönlichen Fähigkeit des Unternehmensleiters/der Leitungsgruppe, die Einführung von EDV als strategische Aufgabe zu behandeln und die dazu notwendigen Planungsmittel anzuwenden (Die notwendige intensive Schulung mit Vermittlung einer Arbeitsmethodik wird derzeit nur von wenigen Anbietern, z.B. Helfrecht oder Schmidt Kolleg, geboten und nur von einem sehr kleinen Anteil der Betriebe genutzt.);
- dv-kundige Betriebsberater. Da jedoch die Honorarsätze bei Einsätzen für Klein- und Mittelbetriebe (auch bei Förderung) beschränkt und die Einsatzdauer limitiert ist, versuchen höher qualifizierte Berater für zahlungskräftigere Kunden in größeren Unternehmen zu arbeiten. Im Mittel zeigen daher zum Einsatz kommende Berater ein schwaches DV-Profil;
- die Softwareanbieter, die in den meisten Fällen örtliche oder regionale Betriebe sind, oft sehr kleine Betriebe, die entweder †Standardsoftware eines überregionalen Herstellers oder eigene Lösungen anbieten;
- die Qualität dieser Softwareanbieter (sie ist sehr unterschiedlich, oft an der professionellen Untergrenze.), ebenso viele angebotene Produkte;
- Standardsoftware, die für viele Einsatzbereiche bei typischen Dienstleistern reichlich vorhanden ist, z.B. für Anwaltskanzleien, Makler, Friedhofsgärtner, Hotels etc.

Es ist aber außerordentlich schwer in diesem Angebot auszuwählen, da selten Vergleiche konkurrierender Produkte durchgeführt werden, die Neutralität und die Kriterien der Vergleiche öfters zweifelhaft sind, wissenschaftlich basierte Vergleiche in der Regel so kompliziert sind, daß sie für die Betriebsinhaber zu komplex und in der entscheidungsbezogenen Durchführung zu teuer sind.

Standardsoftware für kombinierte Dienstleister/Fertiger/Händler ist kaum erhältlich - ausgenommen für den kaufmännischen Bereich. Dies bedeutet, daß für solche Betriebe mehrere Standardsoftware betriebsbezogen integriert werden müssen, bzw. daß †Individualsoftware erstellt werden muß (die dann ggf. vom Anbieter zu †Standardsoftware weiter entwickelt wird). Entsprechend ist bei dieser Gruppe der "Kombinierten" der DV-Einsatz besonders teuer und - zumindest bzgl. integrierter Lösungen - noch kaum in Angriff genommen.

Die Folgen der Engpässe

Häufig sind folgende negativen Ergebnisse aufgetreten:

- innerhalb von 5-10 Jahren mehrere Versuche mit jeweils neuen Lösungen, die allesamt gescheitert sind bzw. die nach hohen Kosten schließlich doch noch zum Ziel geführt haben;
- Verlust guter Mitarbeiter, die wegen Akzeptanzproblemen mit der eingeführten DV-Lösung das Unternehmen verlassen oder dies beabsichtigen;
- Arbeit mit einem Anwendungssystem, das kaum oder nur bruchstückhaft genutzt wird;
- Betrieb mehrerer Systeme mit Medienbrüchen, da die Integration nicht gelöst ist.

Diese negativen Ergebnisse werden meist von den Betroffenen verschwiegen, oft auch gar nicht in vollem Umfang erkannt.

Erfolgreiche DV-Einführungen

Wirklich erfolgreiche DV-Einführungen bei dienstleistenden Klein- und Mittelbetrieben sind bisher vergleichsweise selten. Am ehesten gelingt dies wohl, wenn alle günstigen Faktoren zusammentreffen: eine strategisch qualifizierte Betriebsleitung, ein fachlich einwandfreier Berater und ein solider DV-Anbieter.

Beratungsbedarf

Das angesprochene Unternehmenssegment hat extremen Beratungsbedarf, der bisher kaum befriedigt wird. Mit mittleren und großen Unternehmen erfahrene Berater mit DV-Kenntnissen sollten im Regelfall nicht eingesetzt werden, da die Kenntnisse meist nicht übertragbar sind. Ein wichtiger Weg zur Schaffung geeigneter Beratungskapazität wird die Bereitstellung rechnergestützter Beratungshilfen sein, die leicht und billig einsetzbar sind und als Nebenprodukt auch Betriebsvergleiche liefern. Vermutlich nur auf diesem Weg ist

- die große Zahl der kleinen Dienstleister/Klein- und Mittelbetriebe überhaupt mit zu bedienen und
- zum niedrigen Kostenrahmen ein qualifiziertes Ergebnis zu erbringen.

Das Erstellen der Strukturen für solche Beratungssysteme und die Realisierung der Beratungssysteme ist eine dringend anstehende Aufgabe für Forschung und Entwicklung.

Geeignete Softwarestrukturen

Der Großteil der bestehenden Anwendungsprogramme ist mit den ↑Programmiersprachen etc. entwickelt worden, die die Entwickler jeweils bevorzugten, oft (fast) ohne Werkzeug. Viele dieser Produkte laufen noch auf der MS DOS-Basis, integrieren keine Textverarbeitung, etc. Anders als im Bereich größerer Unternehmen ist bei den Softwareanbietern oft das Bewußtsein für den Einsatz von Werkzeugen und ↑Datenbanken für erhöhte Produktqualität und Produktivität noch wenig ausgeprägt.

Zur Zeit haben progressive Softwareanbieter für den Bereich der Klein- und Mittelbetriebe damit begonnen, qualifizierte ↑Middleware-Plattformen und/oder objektorientierte Werkzeugkästen einzuführen. Diese Entwicklung steht jedoch noch am Anfang, erste Berichte sind - auch wirtschaftlich - ermutigend.

Neue Beratungsqualität

Es ist anzustreben, daß sich in den nächsten Jahren eine neue Beratungsqualität mit den genannten Eigenschaften entwickelt. Hier trifft die Hochschulen in den Bereichen Betriebswirtschaft und ↑Wirtschaftsinformatik eine besondere Aufgabe. Es gilt, vom bisher fast aus-

schließlichen Schwerpunkt der mittleren und großen Unternehmen und ihrer Strukturen abzurücken und den riesigen Bedarf an qualifizierten Beiträgen im Bereich der Dienstleister/Klein- und Mittelbetriebe besser zu befriedigen.

Prof. Dr. Peschanel, Ilmenau

Arbeitsplatzrechnungssysteme

Systeme - auch ↑Workstations genannt - die zunehmend eine hohe Rechenleistung und einen großen Datendurchsatz mit hochauflösender Graphik kombinieren und dem Benutzer vor Ort für eine Vielzahl von Datenverarbeitungsanwendungen zur Verfügung stellen. Darüber hinaus besitzen sie meistens einen Zugang zu Kommunikationsnetzwerken, mit deren Hilfe sie auch auf weltweit verfügbare Informationsquellen zugreifen können. Damit sind Workstations auch Einrichtungen zur Kommunikation und zum Datenaustausch mit anderen Benutzern. Diese Systeme verfügen vermehrt auch über audiovisuelle Ein- und Ausgabemöglichkeiten für eine Bearbeitung von Multimedia-Daten, für einen Austausch von Stand- und Bewegtbildern ebenso, wie für die Übertragung akustischer Informationen.

Architektur betrieblicher**Metainformationssysteme**

↑Metainformationssysteme, Architektur betrieblicher

Architektur Offener Systeme

↑Offener Systeme, Architektur

Architektur von Datenbanksystemen

↑Datenbanksystemen, Architektur von

Archivierung

Auslagerung von Dokumenten auf externe Datenspeicher (elektronische Ablage).

ARPA-Net

Advanced Research Projects Agency Network; Fernnetz mit folgenden Merkmalen: Netzbetreiber und Netznutzer sind i.a. verschieden; es können verschiedene Arten von Informationen, darunter Dokumenten, Daten, Programmen als Poststücke übertragen werden; der Zugang zu verschiedenen Informationsbanken und zu entfernten Ressourcen (Remote Job Entry) ist möglich.

Arrays

Feste Anzahl von Datenelementen, die hintereinander abgespeichert werden und auf die mittels Indizes zugegriffen wird.

Artificial Reality

Synonym zu Virtual Reality, Virtual Environment; eine neue Epoche in der Mensch-Maschine-Kommunikation, die mit dem Einsatz innovativer Endgeräte den Benutzer in vier- und mehrdimensionale rechnerinterne Modelle einbezieht; bezogen auf die menschliche Sensorik Einsatz multimedialer Präsentationsformen, sowie Einbeziehung neuer, multidimensionaler Interaktionstechniken.

ASCII

American Standard Code of Information Interchange

Assembler

Maschinenorientierte Programmiersprache; lehnt sich eng an die interne Logik und Speichertechnik des Computers an und zeichnet sich durch präzise Formulierung des Lösungsweges aus, wozu ein Befehlsvorrat von über 140 Einzelinstruktionen zur Verfügung steht.

Assoziation

Beziehungsmenge zwischen Klassen; ist inhärent bidirektional und modelliert die Menge der Beziehungen zwischen einzelnen Objekten der beteiligten Klassen.

AT-Technologie

Advanced Technology

Aufgabenanalyse im Büro

↑Büro, Aufgabenanalyse im

Aus- und Weiterbildungssysteme,**computergestützte***Problemstellung*

Seit geraumer Zeit ist eine immer rasantere, für viele geradezu beängstigende Vermehrung von verfügbarem Fachwissen zu beobachten. Parallel dazu steigen in vielen Bereichen die beruflichen Qualifikationsanforderungen kontinuierlich. Ein immer breiter und an vielen Stellen auch tiefer angelegtes ↑Wissen muß einem immer größeren Personenkreis vermittelt werden. An den Bildungseinrichtungen, insb. an den Hochschulen, wird deshalb oft von einer Massenausbildung gesprochen, bei der auf individuelle Be-

dürfnisse kaum Rücksicht genommen werden kann. Im Beruf ist eine betriebliche Weiterbildung "auf Vorrat" häufig nicht mehr sehr effizient und insb. unökonomisch. Das geforderte lebenslange Lernen ist flexibler zu gestalten. Es muß immer häufiger ad hoc Zusatz- oder Spezialwissen für die Bewältigung anstehender Probleme parallel zur täglichen Arbeit erworben werden.

Nachdem Computer an Schulen und Hochschulen, sowie im privaten und insb. im betrieblichen Bereich immer weiter verbreitet sind, liegt es nahe, dieses Werkzeug auch für Lehr- und Lernzwecke zu nutzen. Dies ist ohnehin unumgänglich, wenn es z.B. um das praxisbezogene Erlernen des Umgangs mit ↑Anwendungssoftware geht. Darüber hinaus kann der Computer als flexibles Instrument für nahezu beliebige Ausbildungszwecke eingesetzt werden. Eine seiner attraktivsten Eigenschaften ist dabei die Fähigkeit, mit dem Lernenden zu interagieren, d.h. Fehler zu erkennen, zu analysieren, zu beraten, zu verbessern, zu bewerten und zu vergleichen. Neben dem unmittelbaren Feedback hat der Lernende den Vorteil, Zeitpunkt, Zeitdauer und Tempo des Unterrichts selbst bestimmen zu können. Bei Verwendung von mobilen Rechnern wie Laptops oder Notebooks kommt zur zeitlichen auch die örtliche Unabhängigkeit hinzu.

In vielen Fällen wird die Aus- und Weiterbildungssoftware konventionelle Unterrichtsformen nicht völlig ersetzen, sondern Frontalunterricht, Seminare oder Workshops ergänzen und durch Übungs- und Trainingsmöglichkeiten anreichern.

Begriffswelt

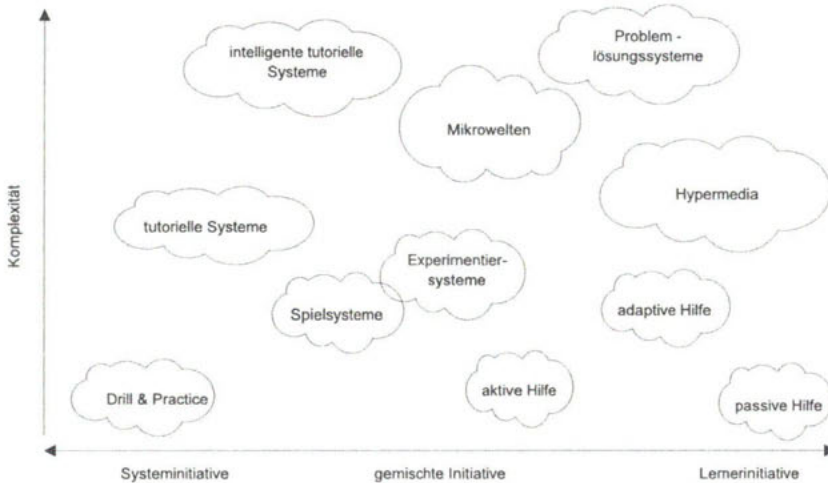
In der betrieblichen Praxis hat sich für das Gebiet des computergestützten Lernens der Begriff ↑Computer Based Training (CBT) eingebürgert. Bei allgemeinen oder wissenschaftlichen Betrachtungen wird im englischen Sprachgebrauch häufig die Bezeichnung ↑Computer Aided Instruction (CAI) verwendet. Durch Kombination der Bausteine "Aided" bzw. "Based" und "Instruction" bzw. "Learning" bzw. "Training" entstehen weitere Abkürzungen (wie CAI, CAL, CAT, CBI, CBL, CBT), ohne daß jedoch wesentliche inhaltliche Differenzierungen vorgenommen werden.

Unter dem Begriff †Computer Managed Instruction (CMI) werden Hilfsmittel für den Lehrenden zusammengefaßt. Hierzu gehören z.B. Systeme zur Lehrstoffverwaltung, Kursorganisation, Prüfungsstellung und -auswertung. Spezielle Werkzeuge zur Entwicklung von Software für die Aus- und Weiterbildung sind Autoresysteme und Autorensprachen.

Methodische Ansätze

Wird die Rolle des Lernenden im Dialog mit einem Lehrsystem betrachtet, so können drei Interaktionsstile unterschieden werden: systemgesteuerter, lernergesteuerter und gemischt-initiativer Dialog.

In der ersten Variante sind die Dialogschritte im wesentlichen vom System vorgezeichnet. Der Lernende nimmt in einer passiven Rolle Informationen auf und reagiert überwiegend auf Anfragen und Vorgaben des Systems. Im spiegelbildlichen Fall agiert der Lernende und kann sich relativ frei nach seinen Vorstellungen im System bewegen bzw. dieses manipulieren. Der Computer reagiert auf gezielte Eingaben oder Befehle. Bei dem gemischt-initiativen Lerndialog übernehmen Lernender und System jeweils aktive wie auch reaktive Rollen.



Lernergesteuerte Lehrsoftware

Die einfachste Art der Wissensvermittlung besteht darin, daß sich der Lernende die benötigten Informationen selbst gezielt abrufen. Konventionelle Hilfesysteme werden z.B. durch eine Anfrage aktiviert und selektieren aus einem online-Manual oder aus einer †Datenbank Informationsbausteine. Flexiblere und individuellere Erklärungen werden von dynamischen Hilfesystemen erhalten, die einerseits den aktuellen Kontext und den bisherigen Dialogablauf berücksichtigen und andererseits aktiv benutzerangepaßte Informationen in den Dialog einbringen.

Wenn sich der Lernende selbständig in einem Informationsnetzwerk bewegen kann, wird von der Kategorie der †Hypermediasysteme gesprochen. Die †Objekte in diesem Netzwerk können Textabschnitte sein, die z.B. als eine Art von Karteikarten oder Lexikoneinträgen zu begreifen sind. Daneben finden sich in modernen Systemen mehr und mehr multimediale Informationsdarstellungen, wie z.B. anspruchsvolle Graphiken, †Animationen, hochaufgelöste und farbige Bilder, Video- und Audiosequenzen. Ein derartiges System ist als eine Art rechnergestütztes Lexikon mit sehr vielen Quer-

verweisen vorstellbar. Um sich bei der Reise in einem Hypermedianetzwerk zu rechtzufinden, bieten derartige Systeme in der Regel Navigations- und Orientierungshilfen an.

Systemgesteuerte Lehrsoftware

Einfache Formen des streng systemgeführten Dialogs sind sog. Drill and Practice-Programme. Das System wählt dabei aus einem Katalog von Aufgaben bzw. Fragen nach einer bestimmten Methodik ein Element aus. Nach seiner Antwort erhält der Lernende unmittelbar ein Feedback. Meist besteht dieses aus der kurzen Bestätigung einer richtigen oder der Verbesserung einer falschen Antwort. Die richtige bzw. fehlerhafte Reaktion beeinflusst dabei den weiteren Prozeß der Fragen- bzw. Aufgabenselektion. Derartige Übungssysteme verfolgen in extremer Form das Ziel, durch gesteuerte Wiederholungen ↑Wissen einzupauken, bis es "sitzt". Bei Testsystemen hat der Lernende ein komplexeres Problem, oder eine zusammenhängende Aufgabenserie zu bewältigen und das System beurteilt anschließend die erbrachte Leistung.

Die wichtigste Form des systemgesteuerten Lernens sind die tutoriellen Systeme. Der Lernende wird dabei auf einem bestimmten Weg durch den Lehrstoff geführt. Um ein individuelles Lerntempo zu ermöglichen, wird der Stoff in einzelne, möglichst kleine Lerneinheiten untergliedert. Diese sind zu einer logischen Abfolge miteinander verbunden, wobei der Schritt zur nächsten Einheit oft von den Antworten in den vorangegangenen Lernsequenzen abhängt. Ein wesentliches Merkmal eines tutoriellen Systems ist, wie oft und in welcher Form eine Lernerfolgskontrolle stattfindet. Es werden Fragen mit Auswahl- oder Zuordnungsantworten, sowie Fragen mit freien Antworten und Zwischentests unterschieden, die mehrere Lösungsschritte beinhalten. Ein Großteil der gegenwärtig verfügbaren Lehrsoftware beruht auf diesem Konzept. In neuerer Zeit wird versucht, mit Hilfe wissenschaftlicher Techniken intelligente tutorielle Systeme (ITS) zu entwickeln. Diese sollen sich generativ und adaptiv verhalten. Mit generativ ist gemeint, daß das System nicht vorgefertigte Lerneinheiten in einer vorstrukturierten Reihen-

folge abspult, sondern Lehrelement und Dialogschritte flexibel erzeugt bzw. zusammenstellt. Eine adaptive Dialogführung nimmt auf die individuellen Bedürfnisse des Lernenden Rücksicht und paßt die Systemausgaben an Vorkenntnisse, aktuelle Verhaltensweise, Lernziele usw. des einzelnen Benutzers an. Entsprechende ↑Informationen und Einschätzungen werden in einem Benutzermodell gehalten.

Gemischt-initiative Lehrsoftware

Die ↑Simulation eines Ausschnitts aus der Realität auf dem Rechner erlaubt in vielen Fällen ein besonders starkes interaktives Lernen. Der Lernende ist oft in ein Szenario integriert und wird zu einem handelnden Bestandteil des Systems. Z.B. hat er in einer Modellfabrik den Part des Qualitätskontrolleurs zu übernehmen und Entscheidungen über Häufigkeit, Art und Umfang der Kontrollen zu treffen. Das System simuliert die Fertigung und gibt ein Feedback in Form von Prüfkosten, Ausschubraten und Fertigungsverzögerungen. Die Möglichkeit, entdeckend zu lernen und erworbenes Wissen auszuprobieren, hebt derartige modellorientierte Experimentiersysteme von eher tutoriell orientierten Ansätzen ab. Verhalten sich solche Modelle "intelligent", so wird häufig auch von Mikrowelten gesprochen. Diese können insb. flexibel auf Warum-Fragen reagieren, durch Beobachtung des Lernenden Fehler und Verständnisschwierigkeiten aufdecken und relativ freie Entdeckungsaufgaben stellen. Eine besonders plastische Darstellung von Mikrowelten wird durch Virtual Reality-Systeme möglich. Sog. Cyberspace-Anwendungen erlauben es dem Lernenden, sich in vom Computer erzeugten Scheinwelten so zu bewegen, als ob er sich selbst direkt dorthin begibt. Der Lernende setzt hierzu eine Spezialbrille mit integrierten Displays auf und zieht sensorische Handschuhe oder einen sensorischen Anzug an. Der Computer erzeugt über diese Displays dreidimensionale Bilder und reagiert auf Kopf-, Augen-, Hand- und Fingerbewegungen. Spielsysteme präsentieren ebenfalls anschauliche Lernumgebungen, die auf Modellen basieren. Sie setzen jedoch spezielle Anreize, um sich mit dem Stoff

zu beschäftigen. Diese Anreize bestehen zum einen in einem besonderen Unterhaltungswert, z.B. in Form attraktiver, multimedialer Bildschirmdarstellungen. Darüber hinaus wird versucht, Phantasie und Neugierde zu wecken, sowie Wettkampfsituationen zu konstruieren. Spielsysteme basieren sehr häufig auf Simulationsprogrammen und lassen sich deshalb nicht exakt gegen Experimentier- oder Mikrowelten abgrenzen. Spielsysteme in der computergestützten Aus- und Weiterbildung benutzen das Spiel mit seinen motivierenden Anreizen nur als Verpackung eines didaktischen Konzepts. So wird in Unternehmensspielen auf unterhaltsamer Weise ein Betrieb oder ein herausfordernder Konkurrenzkampf gegen andere Mitspieler geführt, dabei jedoch gezielt betriebswirtschaftliches Fakten/Handlungswissen vermittelt. Modellbasierte Systeme werden auch dazu verwendet, Umgebungen bereitzustellen, die zur selbständigen Problemlösung benötigt werden. So kann von Problemlösungssystemen oder Problemlösungstrainern eine komplexe Aufgabe vorgegeben werden. Um die Aufgabe zu bearbeiten, ist diese in einzelne Bestandteile zu zerlegen und jedes Detail zu untersuchen. Typische Beispiele sind das Finden und Beheben von Gerätefehlern bei der Reparaturausbildung oder das Erlernen medizinischer Diagnosetechniken. Das Problemlösungssystem hat den Lernenden rechtzeitig auf Fehlverhalten aufmerksam zu machen und Hinweise zu geben, wenn er bei der Aufgabe nicht weiterkommt. Für derartige Systeme, die dem Benutzer bei seiner Arbeit "über die Schulter schauen", wird diese Bezeichnung Kritiksysteme verwandt. Sie können bspw. als wissensbasierte Modellkomponenten oder als †Expertensysteme realisiert sein.

Literatur: Bodendorf, F.: Computer in der fachlichen und universitären Ausbildung. Handbuch der Informatik, Bd. 15.1, München, 1990. Bodendorf, F., Hofmann, J. (Hrsg.): Computer in der betrieblichen Weiterbildung. Handbuch der Informatik, Bd. 15.2, München, 1993.

Prof. Dr. F. Bodendorf, Nürnberg

Ausbildung in der

Wirtschaftsinformatik an Hochschulen
Die *Wirtschaftsinformatik* ist als eine

†Angewandte Informatik und als ein interdisziplinäres Fachgebiet zwischen den Wirtschaftswissenschaften und der Informatik angesiedelt. Ihre Ausbildung und Forschung sind in wirtschafts- und sozialwissenschaftlichen Fachbereichen an Hochschulen eingerichtet. Da in der Praxis eine andauernde große Nachfrage nach Absolventen gegeben ist, wurde in den letzten Jahren die *W.* an den Universitäten und Fachhochschulen ausgebaut (Heinrich, Kurbel, 1988; Bischoff, 1990). Obwohl die *W.*, auch als Betriebsinformatik bezeichnet, ein noch relativ junges Fachgebiet darstellt, bietet mittlerweile jede wirtschaftswissenschaftliche Fakultät dieses Fach an. Die ersten Ansätze lassen sich bereits Ende der 50er Jahre lokalisieren, als an deutschen Universitäten vereinzelt Datenverarbeitungsaspekte in Lehrveranstaltungen über Organisationslehre oder Unternehmensforschung (†Operations Research) aufgenommen wurden. Lehrstühle mit *W.*-Ausrichtung wurden im deutschsprachigen Raum erstmals 1968 an der Hochschule in Linz (Österreich) und 1970 an der Universität in Erlangen-Nürnberg eingerichtet. Die Fachhochschulen haben schon Mitte der 60er Jahre organisatorische Einheiten in *W.* gebildet und 1970 den ersten Studiengang eingerichtet.

Bereits Mitte der 70er Jahre wurde von der wissenschaftlichen Kommission Betriebsinformatik im Verband der Hochschullehrer für Betriebswirtschaft eine Befragung zur Ausbildung an deutschsprachigen Hochschulen durchgeführt, die bis heute mehrmals wiederholt wurde. Die frühe Untersuchung hat ergeben, daß damals grob vier Ausbildungsalternativen angeboten wurden, die sich durch die Anzahl der Semesterwochenstunden und durch den Stellenwert des Fachs innerhalb der Studienordnung unterschieden (als freiwillige Vorlesungen, als Wahlfach, als Vertiefungsfach, als spezielle Betriebswirtschaftslehre). Heute hat sich die *W.* zu einem selbständigen Fach entwickelt, das von einem Lehrstuhl zumindest als Wahlfach angeboten wird. An mehreren Hochschulen wurde die *W.* zu einem selbständigen Studiengang ausgebaut, wobei auch der akademische Grad eines *Diplom-Wirtschaftsinformati-*

kers erworben werden kann. Auch das Ausbildungspotential an Fachhochschulen ist beträchtlich erweitert worden. Viele Fachhochschulen bieten ebenfalls spezielle Studiengänge der *W. an* und verleihen entsprechend den akademischen Grad *Diplom-Wirtschaftsinformatiker (FH)*, aber auch in Informatik-Studiengängen den Grad *Diplom-Informatiker (FH)*, *Studiengang Wirtschaftsinformatik*.

In der Lehre sind deutliche Unterschiede zwischen Grund- und Hauptstudium zu erkennen. Im *Grundstudium* dominieren häufig noch die technischen und instrumentellen Inhalte, d.h. das Kennenlernen des Aufbaus und der Funktionsweise der Hard- und Softwarekomponenten (Hansen, 1992; Stahlknecht, 1993) und/oder das Erlernen einer Programmiersprache, wie z.B. \uparrow PASCAL, MODULA 2, \uparrow C oder \uparrow COBOL, sowie mittlerweile auch logischer und objektorientierter Sprachen (z.B. \uparrow SMALLTALK, \uparrow C++, OBJECT-PASCAL). Ziel des Grundstudiums ist es, den \uparrow Personal Computer mit seinen Betriebssystemen und den wichtigsten Standardsoftwaressystemen kennenzulernen, so die Textverarbeitungs-, Tabellenverarbeitungs- und graphische \uparrow Systeme. Im *Hauptstudium* wird verstärkt auf anwendungsorientierte Veranstaltungen, z.B. über betriebliche \uparrow Informationssysteme, \uparrow Systemanalyse und \uparrow Software Engineering, Datenorganisation und \uparrow Datenbanksysteme, \uparrow Wissensbasierte und \uparrow Expertensysteme, \uparrow Entscheidungsunterstützungs- und \uparrow Managementinformationssysteme (MIS), \uparrow Executive Information Systems (EIS), Bürokommunikationssysteme, sowie Informationsmanagement Wert gelegt.

Die *W.* befaßt sich mit dem Aufbau bzw. der Entwicklung und dem Einsatz bzw. der Nutzung der \uparrow Informations- und Kommunikationssysteme in Wirtschaft und Verwaltung. Entsprechend den späteren Tätigkeitsfeldern der Wirtschaftsinformatiker/innen läßt sich die Hochschulbildung grob in folgende *Ausbildungsbereiche* ordnen:

- \uparrow Anwendungssysteme in Wirtschaft und Verwaltung;
- Entwicklung betrieblicher Anwendungssysteme;

- computergestützte Arbeitsplätze und \uparrow Kommunikation;
- \uparrow Informationsmanagement;
- Informatikmarkt und Menschen;
- Basistechnologie.

Mit *Studienplanempfehlungen* für die Ausbildung in *W.* beschäftigen sich vor allem die wissenschaftlichen Kommissionen Wirtschaftsinformatik im Verband der Hochschullehrer für Betriebswirtschaft e.V., die Schmalenbach-Gesellschaft - Deutsche Gesellschaft für Betriebswirtschaft e.V. und die Gesellschaft für Informatik e.V. Sie erarbeiteten eine gemeinsame Rahmenempfehlung für die Hochschulausbildung *W.*, wobei sie verschiedene *Formen der Ausbildung* unterscheiden:

- *W.*-Ausbildung als Teil eines wirtschafts- und sozialwissenschaftlichen Studiums (insb. der Betriebswirtschaftslehre);
- als Nebenfach für andere Studiengänge (\uparrow Informatik, Mathematik, Ingenieurwissenschaften);
- als eigenständiger Diplomstudien-gang.

Mit dem *W.* sollen grundlegende Prinzipien, \uparrow Methoden, Modelle und Werkzeuge vermittelt werden, die Absolventen befähigen, \uparrow Informations- und Kommunikationstechniken im betrieblichen und überbetrieblichen Rahmen effizient, effektiv und benutzerfreundlich einzusetzen. Das Studium soll berufs- und arbeitsmarktorientiert sein. Die Studieninhalte sollen im Rahmen eines wirtschaftswissenschaftlichen Studiums nach der gemeinsamen Empfehlung in drei Teile gegliedert werden:

- Das *Grundlagenstudium* beschreibt die Breitenausbildung, die alle Studenten/innen durchlaufen sollten (6 SWS), mit folgenden Inhalten \uparrow Computer als Werkzeug zur Problemlösung, Informationsverarbeitung und \uparrow Kommunikation im Betrieb.
- Im *Schwerpunktstudium* sind die Kerninhalte aufgeführt, die die Studierenden bei einer Schwerpunktsetzung erlernen sollten (12-14 SWS): Entwicklung betrieblicher Anwendungs- und Kommunikationssysteme, Softwaretechnologie, Datenorganisation, Informationsmanage-

ment, Anwendungen in ausgewählten Wirtschaftszweigen.

- Das *Vertiefungs- und Spezialisierungsstudium* vermittelt Inhalte, die nach Präferenz der Studierenden und/oder Lehrenden das Schwerpunktstudium punktuell ergänzen können (12 SWS): Technologie der Entscheidungsunterstützungssysteme (↑Expertensysteme, OR-Methoden, Methodenbanken, online-Datenbanken), Informatikmarkt, Computersysteme (Architekturen), Kommunikation (↑Rechnernetze), Anwendungen in ausgewählten Wirtschaftszweigen (Handel, Banken, Verkehr u.a.).

Für Diplom-Studiengänge *W.* an Universitäten wurde 1992 eine *Rahmenempfehlung* verabschiedet (Informatik Spektrum, 1992), die auf neun Semester (Regelstudienzeit) mit ca. 160 SWS angelegt ist. Die Empfehlung geht von der Leitlinie aus, daß Lehrveranstaltungen aus Betriebswirtschaftslehre, Informatik und *W.* als Säulen des Studiengangs etwa gleiches Gewicht haben sollten.

Dem Wirtschaftsinformatiker werden sich in den nächsten Jahren weiterhin ausgezeichnete Berufsaussichten eröffnen. Die Praxis verlangt von allen Absolventen mindestens grundlegende *W.*-Kenntnisse, die auch von allen Hochschulen angeboten werden. Weiterhin bestehen Möglichkeiten der Schwerpunktbildung in *W.* bzw. der Vertiefung oder Spezialisierung.

Literatur: Bischoff, R.: Wirtschaftsinformatik an Fachhochschulen, Studium, Angewandte Forschung und Transfer, Berlin 1990. Heinrich, L.J., Kurbel, K.: Studium und Forschungsführer Wirtschaftsinformatik, Berlin, 1988.

Prof. Dr. R. Gabriel, Bochum

Ausbildung von Ingenieurstudenten in der Informatik

Die Ausbildung von Ingenieurstudenten im Fach ↑Informatik bildet heute einen festen Bestandteil des Studienprogramms aller Bildungseinrichtungen in der Bundesrepublik Deutschland, an denen Absolventen technischer Fachrichtungen die Berufsbezeichnung Diplom-Ingenieur erwerben können. Dazu gehören in erster Linie universitäre Einrichtungen, wie Universitäten, Technische Universitäten und Hochschulen sowie Gesamthoch-

schulen (Dipl.-Ing.) und die Fachhochschulen (Dipl.-Ing. (FH)). In einigen Bundesländern existiert ein solcher Bildungsweg auch an Berufsakademien (Dipl.-Ing. (BA)). Umfang und Inhalt der *I.* differieren dabei nicht nur zwischen diesen drei Plattformen, sondern können durchaus zwischen verschiedenen Einrichtungen einer dieser Plattformen unterschiedlich sein. Unterschiede sind vor allem auch zwischen den einzelnen technischen Studiengängen und sogar Studienrichtungen vorhanden. Verbindliche Regelungen darüber werden in den jeweiligen Studienordnungen getroffen, die auf der Grundlage der Diplomprüfungsordnungen der betreffenden Studiengänge an jeder Bildungseinrichtung erlassen sind. Die *I.* für Ingenieure kann in drei Niveaus unterteilt werden:

- Das erste Niveau umfaßt die sog. *Grundlagenausbildung*, die zu jedem Ingenieurstudium gehört und in die Phase des Grundstudiums eingeordnet ist. Die darin vermittelten Kenntnisse sind so dimensioniert, daß sie einen Bildungsstand garantieren, der sich der allgemeinen Zielsetzung des Vordiploms unterordnet und z.B. bei universitären Einrichtungen einen problemlosen Wechsel der Hochschule ermöglicht.

- Das zweite Niveau (*Vertiefungsausbildung*) ist dadurch gekennzeichnet, daß über die Grundlagenausbildung hinausgehende Informatiklehrveranstaltungen belegt werden. Zu diesem Zweck bieten die Studienpläne einiger Studienrichtungen Informatikdisziplinen als Wahlpflichtfächer an. Vorwiegend handelt es sich dabei um Disziplinen der Praktischen und ↑Angewandten Informatik. An einigen Hochschulen kann dieses Niveau auch durch den Erwerb eines speziellen Zertifikats erreicht werden. Ein Beispiel dafür ist das CIM-Zertifikat, das von einigen Einrichtungen, die über ein CIM-Technologie-Transferzentrum verfügen, ausgestellt wird.

- Das dritte und höchste Niveau betrifft die *I.* in *speziellen ingenieurwissenschaftlichen Studienrichtungen*, die durch ihre Nähe zur ↑Informatik den Erwerb diesbezüglichen Spezialwissens zur Notwendigkeit werden lassen. Dazu gehören insb. Mikroelektronik, Nachrichten- und Automatisierungstechnik.

Grundlagenausbildung, Zielsetzung

Die Grundlagenausbildung verfolgt das Ziel, die Informatik als Wissenschaftsdisziplin mit ausgeprägten struktur- und ingenieurwissenschaftlichen Zügen einzuführen, ihre Bedeutung als Wachstumsfaktor und unverzichtbares Instrument zur Beherrschung der informationellen Prozesse in Wissenschaft, Technik und Produktion bewußt zu machen und theoretische Konzepte, Mittel und Methoden in einem solchen Umfang darzubieten, der die erforderlichen Voraussetzungen für die Anwendung im Beruf schafft. Da die Informatik als Wissenschaft von der systematischen Darstellung und Verarbeitung von Informationen mit Hilfe des Computers definiert ist, geht es vor allem darum, daß die Studenten befähigt werden, Vorzüge und Möglichkeiten seines Einsatzes, aber auch die dafür notwendigen Erfordernisse zu erkennen und mit ihm umzugehen. Umgang mit dem Computer ist dabei unter zwei Aspekten zu sehen. Einerseits muß erreicht werden, daß die in den Bildungsstufen vor dem Studium erworbenen und z.T. erheblichen Unterschiede in bezug auf das Computerverständnis (Computer Literacy) auf einen Mindeststand gebracht wird. Kenntnisse über den Computer und dessen Verwendungsmöglichkeiten sind selbst unter Abiturienten recht unterschiedlich ausgeprägt, insb. was Gründlichkeit, aktives und passives Wissen und Nutzungsniveau betrifft. Andererseits müssen die Studenten in die Lage versetzt werden, informationsverarbeitende Prozesse allgemeinen aber zunehmend berufsspezifischen Inhalts zu analysieren, organisatorisch und algorithmisch aufzubereiten und programmtechnisch zu implementieren. In einem Mindestumfang sind dazu Kenntnisse über Gesetzmäßigkeiten, Wirkprinzipien, Strukturen und Funktionen informationeller Prozesse und Systeme notwendig. Ein noch nicht selbstverständliches Bildungsziel, aber eines mit zunehmender Bedeutung ist es, die gesellschaftlichen Gefahren, die mit einem verantwortungslosen Computereinsatz verbunden sind, bewußt zu machen (soziale Folgen, Datenschutz, militärischer Mißbrauch etc.). Ein eindringlicher Mahner in dieser

Richtung ist der amerikanische Computewissenschaftler (Weizenbaum, 1993).

Problemstellung

Die Verwirklichung der genannten Ziele verdichtet sich auf die Behandlung folgender Probleme:

- Gegenstand, Rolle und Verantwortung der Informatik;
- Aufbau und Arbeitsweise des Computers, Einheit von Hard- und Software, der Computer im Computernetz;
- interaktive Arbeit am Computer und Nutzung problemorientierter Softwaresysteme;
- systematisches Erstellen von Programmen von der Problemstellung bis zu fertigen Programmen in einer höheren Programmiersprache;
- Grundprinzipien der Softwaretechnik, die bei der Entwicklung Anwenderprogramme benötigt werden;
- Methodik des Datenbankentwurfs und Nutzung von Datenbanksystemen;
- diverse Programmierparadigmen.

Die Stoffvermittlung in Form von Vorlesungen, ggf. auch in seminaristischer Form, geht einher mit dem Erwerb praktischer Fähigkeiten durch die Studenten. Grundlage dafür sind i.d.R. vorlesungsbegleitende Übungen, Praktika und das Selbststudium. Diese Fähigkeiten sind:

- die Arbeit am Computer (meist PC, seltener Workstation) in wenigstens einer konkreten Betriebssystemumgebung;
- die Entwicklung einfacher und modular strukturierter Programme unter Verwendung einer höheren Programmiersprache und vorhandener Programmbibliotheken;
- das Erfassen, Testen und Korrigieren von Programmen mit Bezugnahme auf die angebotenen Nutzerschnittstellen;
- die Nutzung vorhandener Anwenderprogramme und Softwaresysteme;
- der Entwurf relationaler Datenbankschemata für ein konkretes Datenbanksystem.

Insgesamt ist die Ausbildung darauf gerichtet, daß sich die Studenten einen korrekten und systematischen Arbeitsstil für einzelne Phasen des Softwarelebenszyklus aneignen und eine disziplinierte, verantwortungsbewußte und effiziente Arbeitsweise entwickeln.

Inhalte

Wegen der eingangs erwähnten Unterschiedlichkeit in der 1. an den einzelnen Hochschulen und in den verschiedenen Studiengängen gibt es auch in der Grundlagenausbildung keine einheitlichen Lehrprogramme zur Behandlung der aufgelisteten Probleme. In der Regel werden die Inhalte der Lehrprogramme zwischen den Informatik vertretenden Bereichen und den jeweiligen ingenieurwissenschaftlichen Bereichen abgestimmt. Nicht selten gehören ingenieurwissenschaftlichen Bereichen Informatiker an, woraus sich eine gewisse Autarkie dieser Bereiche hinsichtlich der Grundlagenausbildung ableitet. Aus diesem Grunde sollen im folgenden stellvertretend zwei

Lehrprogramme lediglich als Beispiele für die Studiengänge Maschinenbau und Elektrotechnik dargelegt werden.

Lehrprogramm: Grundlagen der Informatik für den Studiengang Maschinenbau. Die Lehrveranstaltung umfaßt insgesamt 7 Semesterwochenstunden und ist auf 2 Semester verteilt:

- 1. Semester: 2 Stunden Vorlesung, 1 Stunde Übung wöchentlich.
- 2. Semester: 2 Stunden Vorlesung, 2 Stunden Übung, wöchentlich.

Hinzukommen im Rahmen des Selbststudiums Lehrveranstaltungsbegleitende praktische Übungen am Computer. Die thematische Stundenaufteilung geht aus der folgenden Übersicht hervor.

Thema	Vorlesung	Übung
1. Gegenstand und Nutzen der Informatik, Computer, Betriebssysteme	2	1
2. Überblick über die Programmiersprache PASCAL: einfache und strukturierte Datentypen, einfache und strukturierte Anweisungen	4	3
3. Unterprogrammtechnik: Prozedur- und Funktionsabstraktionen, Rekursivität	2	2
4. Top down-Entwurf von Algorithmen: Phasen des Programmentwurfs, strukturiertes Entwerfen, Entwurfshilfsmittel, iterative und rekursive Entwurfsverfahren	5	1
5. Dateiarbeit: Dateiorganisationsformen, Operationen auf Dateien, Dateiarbeit in Turbo-PASCAL	2	3
6. Dynamische Datenstrukturen: Pointerkonzept, dynamische Variablen, Listen- und Baumstrukturen	3	2
7. Modulare Programme: Modulbegriff, Modultypen und deren Anwendungen, modulares Programmieren in Turbo-PASCAL	3	2
8. Objektorientierte Programmierung: Prinzipien, OO Programmieren mit Turbo-PASCAL	2	3
9. Einführung in die Datenbanktechnologie: Datenmodelle/-schemata, Datenbankentwurf, Sprachkonzepte zur Nutzung von Datenbanken, Schnittstellen zwischen Programmiersprachen und Datenbanken, Entwicklungsrichtungen und Trends	6	4

Lehrprogramm: Grundlagen der Informatik für den Studiengang Elektrotechnik. Die Lehrveranstaltung umfaßt insgesamt 8 SWS und ist auf 2 Semester gleich verteilt mit 2 Stunden Vorlesung, 1 Stunde Übung, 1 Stunde Praktikum wöchentlich. Das Praktikum ist also in die Lehrveranstaltung integriert. Die thematische Stundenaufteilung ist die folgende.

Basismaschine, Methoden und Tools
 Unter der *Basismaschine* wird die Einheit

von Informatik und Systemsoftware (vor allem das Betriebssystem) verstanden. Es wird in die von Neumann-Architektur, die digitale Arbeitsweise der Computer und in die *Schichtenarchitektur* (Computer als System virtueller Maschinen) eingeführt. Als wichtige Schicht wird das Betriebssystem mit so wesentlichen Bestandteilen wie Compiler, Linker, Loader hervorgehoben. Der Vergleich der beiden Lehrprogramme läßt neben formalen Gesichtspunkten hin-

Thema	Vorlesung	Übung
1. Einführung in die Informatik – Gegenstand, Geschichte, Anwendung – Grundlegende Begriffe – Aufbau und Arbeitsweise eines Computers – Funktionen des Betriebssystems – Werkzeuge des Programmentwurfs	6	4
2. Methodik des Programmentwurfs – Darstellungsmittel für Algorithmen und Datenstrukturen – Methoden des systematischen Programmentwurfs – Phasen der Programmentwicklung – Spezifikation, Entwurf, Implementierung, Testung	6	4
3. Programmentwicklung – Struktur von Programmen in C/C++ – Datentypen, statische Datenstrukturen, interne und externe Datenstrukturen – Elementare und strukturierte Anweisungen – Unterprogrammtechnik – Dynamische Datenstrukturen – Klassen – Rekursion – Objektorientierter Entwurf und Programmierung – +-Bibliotheken – Numerische und nichtnumerische Anwendungen	36	20
4. Grundlagen der graphischen Datenverarbeitung	4	4
5. Entwicklung großer Programme – Entwurfskonzepte – Darstellungsmittel und Werkzeuge – Programmierumgebungen	4	-
6. Überblick über andere Programmierungssprachen – Programmierparadigma: imperativ, funktional, logisch; objektorientiert, parallel – Beispielsprachen	4	-

sichtlich der Gliederung (Komplexbildung, Themensequenz) vor allem Unterschiede in der Stoffauswahl und der generellen Auffassung zur Vermittlung einer Programmiermethodik erkennen. Beide Standpunkte und noch andere finden sich in concreto wieder. Das Lehrprogramm für den Maschinenbau baut auf \uparrow PASCAL als Basissprache auf. Fehlende Konzepte, wie Dateiarbeit, modulares und objektorientiertes Programmieren werden im Rahmen der Turbo PASCAL-Erweiterungen ergänzt. Der Schwerpunkt bei der Nutzung kommerzieller Software liegt auf den \uparrow Datenbanksystemen. Hier kommen das 3-Ebenen-Konzept ebenso zur Sprache, wie die drei klassischen Datenmodelle und das Entity Relationmodell. Besonderer Wert wird der Behandlung \uparrow relationaler Datenbanken beigegeben, die auch dem Datenbankentwurf

zugrunde gelegt werden. Dieser Komplex umfaßt ferner die Sprachschnittstellen \uparrow SQL und \uparrow 4GL und moderne Entwicklungskonzepte auf dem Gebiet der \uparrow Datenbanksysteme wie \uparrow verteilte Systeme und Nichtstandard-Datenbanksysteme. Die Stoffauswahl ist begründet, wenn in Betracht gezogen wird, welche wesentliche Rolle in der Vergangenheit klassische \uparrow Programmiersprachen (allen voran \uparrow FORTRAN) bei der Entwicklung von Anwendersoftware für den Maschinenbau gespielt haben, und daß in der Praxis Neuerungen von Nichtinformatikern nur sehr allmählich angenommen werden. \uparrow Datenbanken haben eine herausragende Bedeutung in nahezu allen Bereichen der technischen Produktionsvorbereitung, in der Produktionsprozeßsteuerung und der computerintegrierten Fertigung.

Grundsätzlich kann anstelle von PASCAL auch eine andere imperative Programmiersprache gelehrt werden (ähnlich gut geeignet ist z.B. MODULA-2). Im Falle des Lehrprogramms für die Elektrotechnik ist es C mit der Erweiterung \uparrow C++. Neben dem größeren Sprachumfang, den diese Kombination bietet, erlaubt sie insb. maschinennahe Programmierung (ist dann auch für die Entwicklung von Systemprogrammen geeignet) und die \uparrow objektorientierte Programmierung. Die große Anzahl mittlerweile verfügbarer C- und C++-Programmibliotheken unterstützt den Anwender über ein breites Einsatzspektrum, zu dem wichtige Teilgebiete mit Relevanz für die Elektro- und speziell die Informationstechnik gehören. Auf eines dieser Teilgebiete, die graphische Datenverarbeitung, wird besonders eingegangen. Nach der Vermittlung von Grundkonzepten werden nacheinander die graphische Programmierung in einer C-Umgebung und die Nutzung kommerzieller Graphiksysteme behandelt. In der Verwendung von PASCAL oder C spiegelt sich ein genereller Unterschied hinsichtlich der Programmiermethodik wider. Mit PASCAL wird der traditionelle Weg beschritten, eine aus pädagogischer Sicht besonders geeignete Lehrsprache einzuführen, die einen leichten Übergang zu anderen praxisrelevanten Sprachen erlaubt. Mit C/C++ dagegen wird eine Sprache gelehrt, die sich gegenüber einer „reinen Sprachlehre“ recht widerspenstig verhält, angesichts ihrer zunehmenden Verbreitung in der Industrie und der naturwissenschaftlich-technischen Forschung aber pragmatische Vorteile hat. Weitgehende Übereinstimmung herrscht bei der Behandlung softwaretechnologischer Aspekte. So werden die Phasen des Softwarelebenszyklus, insb. Spezifikation, Entwurf, Implementation und Testung gleichermaßen berücksichtigt, wie die Methode des \uparrow Prototyping. Werkzeuge zur strukturierten Programmentwicklung, wie \uparrow Struktogramm, Programmablaufplan und \uparrow Pseudocode werden vorgestellt und in ihrer Verwendbarkeit diskutiert.

Software

Die Grundlagenausbildung für Ingenieure basiert vorwiegend auf der Nutzung von

PC. Während die Lehrinhalte weitgehend hardwareunabhängig dargeboten werden, kommen PC im Praktikum und beim Selbststudium zum Einsatz. Bevorzugtes \uparrow Betriebssystem ist MS-DOS. Andere Betriebssysteme, insb. UNIX, finden Erwähnung. Die eingesetzten Programmiersysteme richten sich nach der verwendeten Programmiersprache. Bei PASCAL wird gewöhnlich auf die Version Turbo PASCAL 6.0 zurückgegriffen. Geeignete C++-Umgebungen werden z.B. durch Borland C++ ab V.2.00, Zortech C++ Compiler und das GNU-C++System bereitgestellt. Im Komplex Datenbanken werden die Query Language SQL und als Systeme dBase und Oracle bevorzugt. Mit der Verwendung von Informix wird gelegentlich die Schwelle zur UNIX-Umgebung überschritten. Als Applikationssystem werden Produkte zur Textverarbeitung (Word, WordPerfect), \uparrow Tabellenkalkulation (Excel), sowie Zeichen- und Graphikprogramme (Corel Draw, Draw Perfect, PC Draft) angesprochen und mehr oder weniger vertieft.

Vertiefungsausbildung

Die Vertiefungsausbildung ist dem Wesen nach eine Nebenfachausbildung Informatik im Rahmen des ingenieurwissenschaftlichen Studienganges. Wahlpflichtfächer können aus dem eingeschränkten Fächerspektrum des Hauptstudiums des Studienganges Informatik ausgesucht werden. Dazu werden Empfehlungen im jeweiligen Studienführer und durch die Studienberatung gegeben. Besonders geeignet sind Fächer, die echt zur Vertiefung des in der Grundlagenausbildung erworbenen \uparrow Wissens beitragen, wie \uparrow Betriebssysteme, Softwaretechnik, \uparrow Datenbanksysteme, \uparrow Computergraphik, \uparrow Künstliche Intelligenz, Rechnerkommunikation, oder solche der Spezialisierungsrichtung Ingenieurinformatik, die auf Wissenserweiterung im Hinblick auf spezielle Anwendungsgebiete abzielen. Dazu gehören z.B. Gebiete wie Modellierung und Simulation, Geometrische Modellierung, multimediale Techniken, Bilderkennung und -verarbeitung, Robotik. Die realen Möglichkeiten sind vom Profil der jeweiligen Hochschule abhängig.

Spezialausbildung

Bei der Ausbildung von Diplomingenieur-

ren für das Gebiet, das sich mit der Entwicklung informationstechnischer Systeme befaßt, kommt es zu einer engen Verflechtung zwischen Fächern der Informatik und denen der Elektrotechnik. Der Grund dafür ist in der engen Verzahnung dieser Disziplinen (selbst) und in den Überschneidungen bei den Gegenständen zu suchen (Fak, 1991). Die beiden Fakultätentage für Elektrotechnik und Informatik treffen dabei eine Zuordnung von Gebieten, die im Überschneidungsbereich Informationstechnik liegen, zur Informatik bzw. Elektrotechnik. Solche in Studienplänen der Elektrotechnik anzutreffende Informatikdisziplinen sind u.a. Automatentheorie, Betriebssysteme, Informationssysteme, funktioneller Entwurf von Rechnersystemen, Rechnerarchitekturen, Rechnersysteme, Rechnernetze.

Literatur: Fak: Gemeinsame Stellungnahme der Fakultätentage Elektrotechnik und Informatik zur Abstimmung ihrer Fachgebiete im Bereich Informationstechnik, 1991. *Weiz:* Joseph Weizenbaum: Wer erfindet die Computermythen, Freiburg, 1993.

Prof. Dr. H. Schreiter, Chemnitz-Zwickau

Auskunftssysteme

Bereitstellung von in ↑Datenbanken gespeicherte Daten mittels wahlfreiem Zugriff zwecks Informieren von Nutzern. Ihre Anfragen sind in einer Abfragesprache formuliert oder in letzter Zeit auch als natürlichsprachliche Dialoge gestaltet. Beispiele: Auskunftssysteme über Abfahrts-/Ankunftszeiten öffentlicher Verkehrsmittel; elektronische, multimediale Produktkataloge in Waren- oder Versandhäusern; Literaturrecherchesysteme.

Auswahlprozeß von Hardware, Software und Dienstleistungen

Unternehmens- und Informatikstrategie
Technologieorientierte Entscheidungsparameter spielen zunehmend eine wesentliche Rolle in den Unternehmen. Märkte werden schwieriger und Produktlebenszyklen kürzer. Die notwendige, richtige Nutzung der fast explosiv fortschreitenden Informations- und ↑Kommunikationstechnik stellt an das Management veränderte Anforderungen. Konkrete Unternehmensstrategien (Hanker, 1990) - wie Kooperation oder Diversifikation - haben die Ziele Arbeitsteilung, Flexibilität und

Synergiegewinnung, bzw. die Ziele Gewinnung von Produktsynergien (Forschung und Entwicklung) und Marktsynergien (Vertrieb) als Motor. Die daraus ableitbaren Stoßrichtungen der ↑Informatik wären u.a. Standardisierung, Einrichtung eines Informationsverbunds und ggf. eine zentrale Informatikverwaltung und die Verbesserung der Managementkontrolle (↑Decision Support Systems) oder die Notwendigkeit der Unterstützung der Koordination der dezentralen Einheiten (Kommunikationsinfrastruktur).

Eigenproduktion, Fremderstellung, Standardanwendungssoftware

Unter der Zielrichtung dieser genannten strategischen Ziele und informatischen Implikationen sollten ↑Auswahlentscheidungen von Hard- und Software, sowie von Dienstleistungen erfolgen und erst an zweiter Stelle unter produktvergleichenden, produktinternen oder mehr operationalen Gesichtspunkten:

- Bei Existenz eines Informationsverbundes im Sinne einer engen Kooperation dürfte die reine Eigenerstellung von Software oder der Kauf exotischer Hardware nicht der richtige Weg sein: Fremderstellung oder Standardanwendungssoftware verlangen spezielle Anpassungsmechanismen, die vorhanden sein bzw. vorgesehen werden müssen.
- Die Erreichung einer hohen Flexibilität kann durch eine entsprechende Modularisierung gesichert werden. ↑Standardsoftware könnte ggf. flexibler - weil funktional breiter - realisiert sein bzw. leichter dem Bedarf entsprechend parametrisiert werden.

Auslagerungen oder Selbstversorgung

Outsourcing könnte den Ballast von Routinearbeiten reduzieren und die Möglichkeit eröffnen, sich auf informatikstrategische ↑Informationsverarbeitung beschränken zu können, andererseits können Abhängigkeiten entstehen, die unberechenbare Risiken schaffen. Die Verlagerung von Risiken auf externe Partner muß nicht Beseitigung der Risiken heißen. ↑Outsourcing kann leicht zu einer Kostenreduzierungsstrategie werden. Ein DTP-Oursourcing (Auswahl, Installation, Netzentwurf, Support, Wartung, Entwurf neuer Lösungen) für ↑PC und ↑Workstations kann, bei Belassen der Verantwort-

tung der Kernbereiche im Unternehmen, ungefährlich sein.

Entscheidungsfaktoren zur Auswahl von Hard-, Software und Dienstleistungen

Die zentralen Entscheidungsfaktoren zur Auswahl (Bischoff, 1990) lassen sich in

- *systembezogene Entscheidungsfaktoren* von Hardware, Systemsoftware und Standardanwendungssoftware mit reinen (statischen) Systemeigenschaften, lastbezogenen Systemeigenschaften, Kosten und Support;
- *unternehmensinterne Entscheidungsfaktoren* mit organisatorischen, finanziellen und personellen Faktoren;
- *unternehmensexterne Entscheidungsfaktoren* mit gesamtwirtschaftlichen Gegebenheiten, gängigen Technik-/Technologieentwicklungen, Konkurrenten (dortiges Know how, dortige Infrastruktur), Kundenwünschen (Image den Kunden gegenüber), Abnehmern (Verflechtung), Herstellern/Lieferanten (Leistungsstand) etc. einteilen. Bezogen auf die Vergabe von umfangreichen Dienstleistungen von Aufgaben aus dem Informatikbereich können obige Faktoren in direkter und analoger Weise genutzt werden.

Am Beispiel des Outsourcing soll dies verdeutlicht werden:

- Softwareanforderungen des Anwenders an den Outsourcinganbieter (insb. Qualitätssicherung und Sicherheitsmanagement);
- Hardwareanforderungen des Anwenders an den Outsourcinganbieter (insb. Verfügbarkeits- und Kapazitätsmanagement);
- Netzanforderung des Anwenders an den Outsourcinganbieter (insb. Netzmanagement und Netzsicherheit);
- Anforderungen an die Qualitätskontrolle (insb. ihre Durchführung beim Outsourcinganbieter);
- Anforderung an die laufende Tageskommunikation;
- Kosten/Nutzen-Übersicht (Outsourcingrechnung);
- Anforderung an die Überprüfbarkeit der Outsourcingleistung (insb. bezogen auf die Sensibilität und die Sicherheitsstufen der Daten).

Der Auswahlprozeß

Der Auswahlprozeß ist ein verkürzter Systementwicklungsprozeß, denn auch

im Outsourcingfall sind systemtechnische Größen zu fixieren. Mit der *Einholung von Angeboten* (Bischoff, 1990) soll beurteilt werden, welche Alternative die beste Lösung darstellt. Die Voraussetzungen hierfür sind eine durchgeführte Systemanalyse und die Entwicklung einer groben Anwendungskonzeption (*Sollkonzeption*), aus der die Mindestanforderungen und die Optimalanforderungen entwickelt werden. Die Aktivitäten zur Ausschreibungserstellung beinhalten die Schaffung von Marktinformationen, die Erstellung des *Anforderungsprofils* in Form eines Pflichtenheftes, die Auswahl der Anbieter, Gespräche mit den Anbietern. Es folgen die Prüfung und die *Beurteilung* der eingegangenen Angebote.

Nach Anwendung der KO-Kriterien erfolgt eine operationalisierte *Aufstellung der Zielhierarchie* bis hin zu einer bewertbaren Kriterienebene. Anschließend erfolgt die differenzierte *Kosten/Nutzen-Bewertung* und ihre Integration zu einer oder wenigen Maßzahlen pro Alternative. Die für diese Vorgehensweise notwendige und vorauszusetzende Angebotsfeinprüfung erfolgt am besten anhand eines aufgabenspezifischen Fragenkatalogs, der dem Pflichtenheft beigelegt werden sollte.

Hilfsmittel und Methoden

Die Steuerung des Auswahlprozesses kann anhand eines eigens dafür vorgesehenen Vorgehensmodells geschehen. Ebenso (oder ergänzend) sind Aktivitätenlisten, objektbezogene Checklisten (Kriterienkataloge, Fragenkataloge etc.) und Formulare vorzusehen (Bischoff, 1990; Frank, 1980; Futh, 1984; Grupp, 1991; Hansen, 1983). Die letztendliche Entscheidung wird im Kontext der entsprechenden Unternehmensstrategie und der abgeleiteten Stoßrichtungen der Informatik anhand einer Wirtschaftlichkeitsbetrachtung erfolgen. Die weitgehende Nutzung sinnvoller und validierter Maßzahlen/Vergleichszahlen kann dabei den Entscheidungsprozeß objektivieren.

Zur Beurteilung von Hardware bieten sich *Mixe* an. Als Mix wird die Zusammenstellung einer fiktiven Standardoperation aus einzelnen Operationen (Befehlsmix) bezeichnet, deren Ausführungszeiten bekannt sind, oder aber ge-

messen werden können. Die Mixe sind meist nur ↑CPU-bezogen einsetzbar. Beim Benchmark-Test wird ein typisches Bearbeitungsprofil auf einer Anlage des gewünschten Typs zum Ablauf gebracht. Hier sind analytische Verfahren einsetzbar, die auf der Warteschlangentheorie beruhen. Ebenso kann die ↑Simulation über spezielle Simulationssprachen zur Modellierung von Rechnersystem, Arbeitslast und Systemverhalten zum Einsatz kommen.

Mit Softwaremonitoren (Meß- und Analyseprogramm und Hardwaremonitoren (messende und auswertende kleine Mikrocomputersysteme) kommen ↑Tools zum Einsatz, die primär zur Engpaßbeseitigung und Teilauswahl genutzt werden können. Die Beurteilung der formalen Qualität von Software kann schließlich über Kennzahlen/Metriken - statisch, dynamisch - unterstützt werden. Zu den hier auch zu erwähnenden Verfahren gehören die eindimensionalen Bewertungsverfahren, wie die statischen und dynamischen Investitionsrechnungen.

Das praktisch wichtigste und verbreitetste Bewertungsverfahren ist die ↑*Nutzwertanalyse* (Bischoff, 1990). Bei der *NWA* wird eine Zielhierarchie Top down bis auf die sog. Kriterienebene heruntergebrochen. Die unterschiedliche Gewichtung der Ziele, Subziele und Kriterien führt zu einer Gesamtpunktzahl (Nutzwertpunktzahl) für jede Alternative und ermöglicht somit den Vergleich und die letztendliche Auswahl. Die Bewertung unterschiedlicher Kriterien im Alternativenvergleich ist in unterschiedlichen Skalen möglich. Zur Wertsynthese ist ihre Zusammenführung in eine einheitliche Skala, meist in eine Punkteskala (Verhältnisskala) empfehlenswert. Die Gewichte können direkt geschätzt werden oder - etwas genauer - über paarweisen Vergleich (Bischoff, 1990). Ein differenziertes Verfahren wäre eine getrennte Ko-

sten/Nutzen-Betrachtung der Investition(en) über ihre Laufzeit, wobei es sich empfehlen würde, mehrere Nutzenkategorien in Abhängigkeit ihrer Realisierungschance einzuführen. Einen interessanten Ansatz könnte schließlich das ↑*Target Costing (Zielkostenmanagement)* darstellen, um zum einen Standardsoftware oder Fremderstellung und Eigenerstellung zu differenzieren, aber ebenso als Verfahren, um alternative Fremdprodukte bzw. Fremderstellungen bzw. Fremdleistungen zu differenzieren.

Rechtliche Problembereiche

Im Rahmen dieser Darstellung kann nur beschränkt auf diese Problematik eingegangen werden. Auf jeden Fall sind

- die Vertragsarten und Vertragsbedingungen (Miete, Kauf, Leasing; Werkvertragsrecht, Dienstleistungsrecht, Personalgestellungsrecht (Bischoff));
- Gütezeichen Software, z.B. vergeben von der Gütegemeinschaft Software e.V. (GGG; Bischoff, 1990);
- die Norm ↑ISO 9000 zur Softwarequalitätssicherung, insb. ISO 9000 Teil 3 im Softwarebereich;
- die EG-Richtlinie (EG-ABI, Nr. L, 122), die den rechtlichen Schutz von Computerprogrammen erweitert, aber auch die Rechte des Käufers klarer faßt; zu beachten.

Literatur: Bischoff, R.: Die Auswahl von Informatikprodukten, in: *Kurbel, K., Strunz, H. (Hrsg.), Handbuch Wirtschaftsinformatik*, Stuttgart, 1990. *Frank, J.:* Standardsoftware, 2. Aufl., Köln, 1980. *Futh, H. et al. (Hrsg.):* Der Computer im Klein- und Mittelbetrieb, München, 1984. *Grupp, B.:* EDV-Pflichtenheft zur Hardware- und Softwareauswahl, Köln, 1991. *Hanker, J.:* Die strategische Bedeutung der Informatik für Organisationen, Stuttgart, 1990. *Hansen, H.R. et al.:* Standardsoftware. Beschaffungspolitik, Berlin, 1983.

Prof. Dr. R. Bischoff, Furtwangen

B-ISDN, Breitband-I SDN

Übermittlung aller Fernmeldedienste in einem einzigen Universalnetz.

Balkendiagramm

B. (Gantt-Diagramme) zeigen die einzelnen Vorgänge in ihrer Position auf dem Zeitstrahl; Darstellung der zeitlichen Relationen der Vorgänge zueinander.

Bank-Controlling

Vor dem Hintergrund des verschärften Wettbewerbsdrucks, wachsender Komplexität und Dynamik der Bankenmärkte, sowie der zunehmenden Internationalisierung des Bankgeschäfts nehmen die Anforderungen an das Bankmanagement und an das verwendete Steuerungsinstrumentarium ständig zu. Als weitgehend unverzichtbar für die Zukunftssicherung gilt daher im Bankgewerbe zunehmend ein modernes Controlling, das ein integriertes Konzept für die ertragsorientierte Banksteuerung ist.

Über den Begriff des *Controlling* gibt es keine einheitliche Auffassung (Horváth, 1990; Richter, 1987; Weber, 1991). Fest steht jedoch, daß *Controlling* nicht einfach mit Kontrolle gleichgesetzt werden darf, da der zugrundeliegende Terminus "to control" weit mehr umfaßt und soviel bedeutet wie "Steuerung, Lenkung und Überwachung". Die somit im Vordergrund des *Controlling* stehende Steuerungsfunktion weist dabei konkret für Banken eine inhaltliche und eine formale Komponente auf. Materiell zeichnet sich ein geschlossenes Konzept des *B.* dadurch aus, daß sowohl die Gesamtbank als auch die einzelnen Geschäftseinheiten bis hin zum einzelnen Geschäft mit Hilfe eines integrierten Konzepts bewußt ertragsorientiert gelenkt werden. Dabei geht es im Kern um die Formulierung und Durchsetzung einer Geschäftspolitik, die ihre Philosophie aus den folgenden drei Grundprinzipien herleitet (Schierenbeck, 1994):

- Es gilt das Primat der Rentabilität! Alle geschäftspolitischen Entscheidungen sind stets der Frage zu unterwerfen, ob bzw. inwiefern sie zur Erzielung einer angemessenen (Mindest-) Rentabilität beitragen. Das *Controlling* fungiert damit als eine Art institutionalisiertes "Ertragswissen" der Bank.

- Geschäftswachstum wird grundsätzlich nicht als Selbstzweck angestrebt, sondern wird ausschließlich als Mittel zur Rentabilitätsmehrung und Rentabilitätsicherung gesehen. Dabei erfolgt stets eine Abstimmung zwischen wachstumsbedingtem Rentabilitätsbedarf und den voraussichtlichen wachstumsbedingten Rentabilitätswirkungen (ertragsorientierte Wachstumspolitik).

- Die Übernahme von Risiken wird dem Rentabilitätsdenken insofern konsequent untergeordnet, als sie sich stets aus den dabei zu erwartenden Ertragsmöglichkeiten zu rechtfertigen hat und strikt mit der generellen Risikotragfähigkeit der Bank gekoppelt sein muß. Das *Controlling* impliziert also eine deutlich defensive Grundhaltung, die das Eingehen von Risiken betrifft (ertragsorientierte Risikopolitik).

Ergänzend zu der materiellen Komponente weist das *B.* stets auch eine formale Komponente auf. Bezogen hierauf kommt dem *Controlling* zum einen die Aufgabe zu, die Rationalität bankbetrieblicher Entscheidungsprozesse durch systematische Planaktivitäten und Erfolgskontrollen sicherzustellen. Zum anderen hat das *Controlling* eine ausgeprägte Koordinations- und Informationsfunktion zu erfüllen. Das *Controlling* kann also institutionell als eine Art *Informationszentrum* verstanden werden, das steuerungsrelevante Informationen erfaßt, aufbereitet und weiterleitet, um die Aktivitäten der einzelnen Geschäftseinheiten im Hinblick auf die Gesamtbankziele zu koordinieren und abzustimmen.

Das Wesen des *B.* kann zusammenfassend durch zwei interdependente Aspekte beschrieben werden: Im materiellen Sinne steht das *B.* für eine integrierte Managementkonzeption, die die betonte Ertragsorientierung zum tragenden Fundament erhebt. Aus formaler Sicht vollzieht sich das *B.* als komplexer kybernetischer Prozeß von revolvierend ablaufenden Planungs- und Kontrollaktivitäten, die in allen Phasen durch systematisches *Informationsmanagement* abgestützt werden. Im Mittelpunkt steht dabei die Formulierung und Abstimmung der (ertragsorientierten) Unternehmensziele auf Gesamtbankebene und die Koordina-

tion aller Einzelaktivitäten und Geschäftsbereiche bzgl. dieser Ziele.

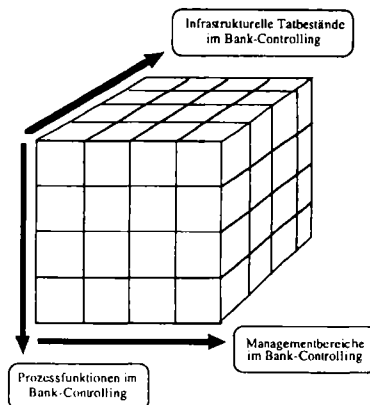
Controllingkonzeptionen in Banken lassen sich grundsätzlich auf eine einheitliche Grundphilosophie zurückführen. Das schließt nicht aus, daß sie in der praktischen Umsetzung differenziert nach institutsindividuellen Rahmenbedingungen zu gestalten sind. Controlling bedeutet nicht ein neues System, das den Unternehmenserfolg automatisch garantiert und entbindet die Entscheidungsträger nicht von den Führungsaufgaben. Vielmehr beinhaltet es die Förderung des Verantwortungs- und Ergebnisbewußtseins der Entscheidungsträger und damit letztlich die Sicherstellung einer konsequent zielgerichteten und - sowohl materiell als auch organisatorisch verankerten - systematischen Entscheidungsfindung auf allen Unternehmensebenen.

Vor dem Hintergrund der so umrissenen Grundphilosophie des *B.* stellt sich dem Management von Banken ein Komplex von Aufgaben, der mit der Durchsetzung des Controllingdenkens in allen Bereichen einer Bank beginnt und bis hin zur Feinsteuerung kleinster Geschäftseinheiten reicht. Die einzelnen Teilaufgaben des *B.* lassen sich dabei von verschiedenen Seiten systematisieren, wobei sich die drei Ebenen des gesamten Aufgabenkomplexes anschaulich in Form eines Würfels darstellen lassen. Die kompakte Darstellung als Aufgabenwürfel weist schon darauf hin, daß es sich bei den einzelnen Dimensionen um ein festes Bündel von Aufgaben handelt, und die Teilaufgaben nicht isoliert betrachtet werden dürfen. Vielmehr existieren komplexe Interdependenzen, die ein integratives Denken und Handeln notwendig machen. So werden bei jedem controllingspezifischen Problem grundsätzlich sämtliche drei Dimensionen angesprochen:

Aufbau und Sicherstellung einer den Grundgedanken ertragsorientierter Banksteuerung adäquaten Infrastruktur. Eine solche Infrastruktur richtet sich auf die Manifestierung des Controllinggedankens auf allen Führungsebenen einer Bank und auf die Gestaltung der Struktur des Managementsystems, d.h. auf die Schaffung der organisatorischen, der planungs- und kontrollrelevanten, sowie der informationsbedingten Voraussetzungen

einer ertragsorientierten Unternehmenssteuerung. Das Kerngerüst einer controllingadäquaten Infrastruktur setzt sich stets aus folgenden vier elementaren Bausteinen zusammen:

- aus der ertragsorientierten Geschäftsphilosophie, die als Kernelement eines integrierten Controllingsystems steht - gleichbedeutend für eine Managementkonzeption, die die betonte Ertragsorientierung zum tragenden Fundament erhebt und dies auch in einer ganz spezifischen Art des Denkens und Handelns auf allen Führungsebenen einer Bank zum Ausdruck bringt. Ertragsorientierung heißt in diesem Sinne, daß das Bankergebnis konsequent im Mittelpunkt geschäftspolitischer Überlegungen steht.



- aus Controllingsystemen, die ihrem Kern nach komplexe Steuerungssysteme sind, bei denen die Planungs- und Kontrollaktivitäten nicht isoliert und unverbunden vollzogen werden, sondern sich konzeptionell in ein komplexes, hierarchisch vermaschtes kybernetisches Regelkreismodell einfügen. Um in diesem Sinne einen *Controllingzyklus* zu institutionalisieren, ist es unerlässlich, strategische und operative Ziele vor dem Hintergrund einer ertragsorientierten Geschäftsphilosophie zu formulieren und in strategische wie auch operative Plangrößen umzusetzen. Die daraus resultierenden Sollgrößen dürfen nicht nur der Gesamtbank als Orientierung dienen, sondern auch den einzelnen Profit-Centern und Kundengeschäften, so daß diese Planaktivitäten zu einem Gesamtplan ko-

ordiniert werden können, in dem sich dann die ertragsorientierte Zielkonzeption der Bank widerspiegelt. Dieser Gesamtplan ist dann die Grundlage und Voraussetzung dafür, Kontrollmaßnahmen zu ergreifen, Probleme, die letztlich Soll-/Istabweichungen darstellen, überhaupt zu erkennen und kurskorrigierende Vorschläge zu unterbreiten.

- aus dem dritten Baustein einer controllingadäquaten Infrastruktur, die die Organisationsstruktur einer Bank bildet, da neben der Durchsetzung einer bestimmten geistigen Grundhaltung auch der konkrete organisatorische Rahmen vorhanden sein muß, in dem die ertragsorientierte Geschäftsphilosophie am Markt umgesetzt werden kann. Hierbei sind vor allem Entscheidungen darüber zu treffen, nach welchen Organisationsprinzipien die einzelnen Unternehmensbereiche zunächst abgegrenzt und später in ihrem Zusammenwirken wieder koordiniert werden und in welchem Maße Entscheidungsbefugnisse an untere Führungsebenen delegiert werden sollen. Eine controllingadäquate Verwirklichung bildet in diesem Zusammenhang die sog. Matrix-Organisation, indem Marktbereiche objektbezogen als kundenorientierte Profit-Center und die zentralen Abteilungen als funktionsorientierte Fachressorts fungieren (Schierenbeck, 1994).

- Als ein weiterer zentraler Baustein für integrierte Controllingsysteme in Banken gilt die Existenz eines steuerungsadäquaten *Führungsinformationssystem*s. Die formale Komponente des Controlling bezieht sich auf die Koordination sämtlicher Unternehmensaktivitäten durch systematisches Informationsmanagement. Da jede Entscheidung nur so gut sein kann, wie ihre informativische Grundlage, ist die Qualität von Bank-Managemententscheidungen auch davon abhängig, inwieweit es gelingt, diese Entscheidungsgrundlagen zu optimieren. ↑Informationen werden so zu einer strategischen Managementressource. Ein ↑Informationssystem soll dabei vor allem der Erfassung, Speicherung und Distribution von relevanten Informationen dienen, d.h. dafür Sorge tragen, daß die benötigten und richtigen Informationen am richtigen Ort zur rechten Zeit zur Verfügung stehen.

- Institutionalisierung eines schrittweisen Vorgehens bei der Wahrnehmung controllingspezifischer Prozeßfunktionen, um so durch die konsequente Einhaltung eines Stufenschemas über Situationsanalyse, Planungs- und Kontrollphase die Geschlossenheit des Controllingzyklus zu garantieren. In all seinen Phasen wird dieser Prozeß durch systematisches Informationsmanagement abgestützt. Da die spezifischen Informationsfunktionen im *B.*, die im einzelnen die Phasen Informationsbeschaffung, -verarbeitung (im Sinne einer Umformung, Verdichtung und Spezifizierung von ↑Informationen), -speicherung sowie -übermittlung umfassen, stets auf das Engste mit den entsprechenden Planungs- und Kontrollfunktionen verbunden sind, werden sie gleichsam implizit in den als komplexen Informationsprozeß gedeuteten Planungs- und Kontrollzyklus integriert. Dessen controllingspezifischen Prozeßphasen beinhalten vornehmlich die Ziel- und Problemanalyse, die Erarbeitung von Entscheidungsvorlagen, sowie Kontrollen und Abweichungsanalysen.

- Die dritte und letzte Dimension des Aufgabenwürfels im *B.* bilden die einzelnen Bankmanagementbereiche ↑Portfolio-, Bilanzstruktur- und Budgetmanagement und deren Moderation nach den Grundsätzen ertragsorientierter Banksteuerung. Gemäß seiner prinzipiellen Stabsfunktion kann das *B.* die einzelnen Managemententscheidungen selbst nicht dirigieren, sondern auf sie nur durch Sachinformationen Einfluß nehmen. Die inhaltlichen Schwerpunkte der Controllingaktivitäten liegen hierbei in der Moderation der Markt- und Wettbewerbspolitik, in der Entwicklung von Vorschlägen zur risikopolitischen Optimierung der Bilanzstruktur und der Planung struktureller Rentabilitätsvorgaben, sowie in der Feinsteuerung der Bankrentabilität durch Budgetierung und Budgetkontrollen.

Literatur: Horváth, P.: Controlling, München, 1990. Richter, H.J.: Theoretische Grundlagen des Controlling, Strukturkriterien für die Entwicklung von Controlling Konzeptionen, Frankfurt a. M., 1987. Schierenbeck, H.: Ertragsorientiertes Bankmanagement, Bern, 1994.

Weber, J.: Einführung in das Controlling, Stuttgart, 1991.

Prof. Dr. H. Schierenbeck, Basel

BASIC

Beginners All Purpose Symbolic Instruction Code; imperative ↑Programmiersprache; Programmstrukturen ähneln den Programmablaufplänen.

Basistechniken der Systemanalyse

↑Systemanalyse, Basistechniken der

BDE (Betriebsdatenerfassung)

Zielsetzung

Mit der *B.* soll der betriebliche Informationsfluß verbessert werden. Dies bedeutet Rationalisierung und Leistungssteigerung zur Sicherung der Wettbewerbsfähigkeit der Betriebe. Insb. unter dem Aspekt zunehmend geforderter Flexibilität der Betriebe sind Möglichkeiten der Beschleunigung des Informationsflusses von eminenter Bedeutung.

Begriffserklärung

B. ist zwar auch das Zusammenfassen, Bewerten und Zurverfügungstellen von Ergebnissen, in erster Linie geht es bei *B.* aber um die Organisation zum Erfassen von Daten, die im Betrieb anfallen. Der Schwerpunkt liegt somit auf dem Erfassen, und zwar vor Ort. Entsprechend lautet die auf (Roschmann, 1979) aufbauende, aktualisiert wiedergegebene Definition wie folgt: *B.* umfaßt die Maßnahmen, die erforderlich sind, um Betriebsdaten insb. eines Produktionsbetriebes in maschinell verarbeitbarer Form am Ort ihrer Verarbeitung bereitzustellen. *B.* ist daher Oberbegriff für eine ganze Reihe von Erfassungsverfahren, wie Maschinen-, mobile, Prozeß-, Qualitäts-, Zeit-, Auftrags-, Lohndatenerfassung etc. Als *Betriebsdaten* sind die im Laufe eines betrieblichen Prozesses, z.B. während des Produktionsprozesses, anfallenden bzw. verwendeten Daten - bei letzteren insb. Nummern zur Identifizierung - zu nennen. Hierbei handelt es sich um technische und organisatorische Daten über das Verhalten bzw. über den Zustand des Betriebes, wie Angaben über produzierte Mengen, benötigte Zeiten, Zustände von Fertigungsanlagen, Lagerbewegungen, Qualitätsmerkmale.

Ein *B.-System* ist ein Hilfsmittel zur Erfassung und Ausgabe betrieblicher Daten

mit Hilfe von automatisch arbeitenden Datengebern (Sensoren) und/oder personell bedienten Datenstationen, sog. *B.-Terminals* im Betriebsgeschehen. Die Erfassungen sind vom *B.-System* vornehmlich dezentral zu unterstützen. Die Datenausgabe erfolgt zunächst zentral mit Orientierung zur auswertenden DV-Anlage, aber auch dezentral, wobei im Betrieb etwas damit angestoßen werden kann (organisatorische und technische Steuerungsinformationen).

Aufgabenstellungen

Die Aufgabenstellung an die *B.* wechselt von Betrieb zu Betrieb. Die Personalzeiterfassung steht in den Betrieben der unterschiedlichsten Branchen einschl. im Dienstleistungsbereich an. Materialdatenerfassungen gibt es in der Industrie, im Handel, bei Speditionen etc. Manchmal treten Aufgaben einzeln auf, in anderen Fällen kombiniert. Bei Produktionsbetrieben haben sich folgende Aufgabenbereiche herauskristallisiert:

- *Produktionsplanung und -steuerung;*
- *Materialwirtschaft* einschl. organisatorischer Lagersteuerung, Transportsteuerung (Materialflußsteuerung, Logistik) mit Vorgaben und Erfassung/Überwachung von Mengenbewegungen;
- *Zeitwirtschaft* mit Termin-/Kapazitätsplanung einschl. kurzfristiger Fertigungssteuerung, Leitstand, Werkstattsteuerung, Arbeitsverteilung, Arbeitsvorgaben und Auftragsrückmeldungen, Auftragsfortschrittserfassung, Zeiterfassung für Auftrags- und Terminüberwachung der Aufträge, sowie Überwachung der Kapazitätsbelegung;
- *Technische Anlagensteuerungen* in leittechnischen Konzepten und in organisatorischen Auftragsablauf eingebettet, wie NC-Maschinen im DNC-Betrieb, integrierte Werkzeugvoreinstellung, Transportsteuerungen u.a.m.;
- *Qualitätssicherung mit Mengenerfassungen* nach gut/schlecht und Erfassung der Daten unterschiedlicher Qualitätsmerkmale gemäß Prüfplanung, ggf. qualitative Einzelobjektverfolgung und Dokumentation einerseits und statistische Qualitätssteuerung andererseits;
- *Betriebliches Rechnungswesen* insb. Kostenrechnung mit ↑Datenerfassung;
- *Personalwesen* mit Datenerfassungen für flexible Arbeits-, Gleitzeit, Lohn-

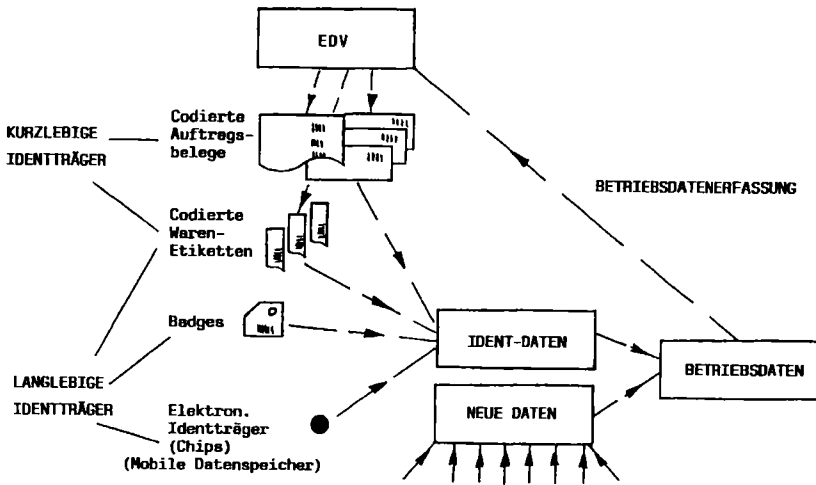
abrechnungen (Leistungsumfang bei Akkord, Prämienlöhnen);

- *Schwachstellenanalysen* z.B. Maschinennutzungsüberwachung zur zeitlichen und ggf. technischen Nutzungserhöhung mit entsprechenden Datenerfassungen (Zeitgrößen, Störgründe, Maschinenprozeßdaten/Prozeßüberwachung);
- *Instandhaltung* mit PPS-ähnlichen Aufgaben, häufig aber schlechteren Plan- und Daten (Beispiel: Reparaturen) - umfassende Datenerfassungsaufgaben, ggf. kombiniert mit Schwachstellenanalysen ↑Diagnosesysteme;
- *Sonstige* wie Wäge-, Tank-, Kantinendatenerfassung, Zutrittskontrolle.

Problemstellung/Inhalte

Betriebsdaten fallen in den Betrieben räumlich verteilt mit den unterschied-

lichsten zeitlichen Bedingungen an. Es gibt stoßweisen Datenanfall und ganz vereinzelt, sowie beide kombiniert. Es sind nicht alle Daten neu, vielmehr müssen eigentlich immer in Ergänzung zu den jeweils neu anfallenden Betriebsdaten auch diese charakterisierenden Daten mit erfasst werden, damit die sachlichen Inhalte feststehen. Die Abbildung zeigt diese Aufgliederung - ergänzt um Beispiele von Identträgern, die die B. unterstützen, indem diese im Erfassungsvorgang automatisch gelesen werden. In der Abbildung sind die Identträger - entsprechend der Kurz- oder Langlebigkeit der Identdaten (Nummern) - aufgefächert. Stark verbreitete Beispiele sind Identträger mit Strichcodes (Barcodes: Virnich, Posten, 1992).

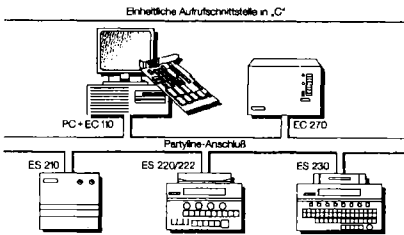


Lösungstechniken/Verfahren

Es stehen heute ausgereifte Techniken und vielfältige Erfahrungen für neue B.-Projekte zur Verfügung. Wissenschaftliche Grundlagenarbeiten mit ihren Ergebnissen, vielfältige anbieterseitige Entwicklungen, Empfehlungen überbetrieblicher Arbeitskreise, Praxiserfahrungen realisierter B.-Projekte haben ein wertvolles B.-Wissen geliefert, z.B. in (Baron et al., 1987) bis (Roschmann, Junghanns, 1993) aufbereitet. B.-Lösungen umfassen ↑Hardware mit gängigen Rechnern, Übertragungswegen und Terminals. Letztere

erfordern dabei die stärksten Anpassungen an die betrieblichen Erfordernisse. Zu den B.-Lösungen werden ferner ↑Software mit systemspezifischen Anteilen benötigt einschl. immer leistungsfähigeren Softwaretools und typischer ↑Anwendungssoftware, die verbreitet mit ↑Tools erstellt, modular strukturiert, vorgefertigt zur Verfügung steht. Dank zusätzlich dokumentierter weiterer ↑Schnittstellen kann dann immer noch projektspezifisch ergänzt werden. Der dritte B.-Lösungsteil umfaßt die betriebliche Organisationskonzeption. Da diese

nicht wie Hard- und Software gekauft werden können, stellt sie für viele Betriebe die zeitintensivste Hürde zur *B.-Nutzung* dar. Auch spielen in diesem Zusammenhang die bei den Mitarbeitern zu erreichende Akzeptanz und der häufig schwierige Nachweis der Wirtschaftlichkeit eine große Rolle (Baron et al., 1987). Das *B.-Angebot* wird jährlich aktuell aufbereitet (Roschmann et al., 1993).

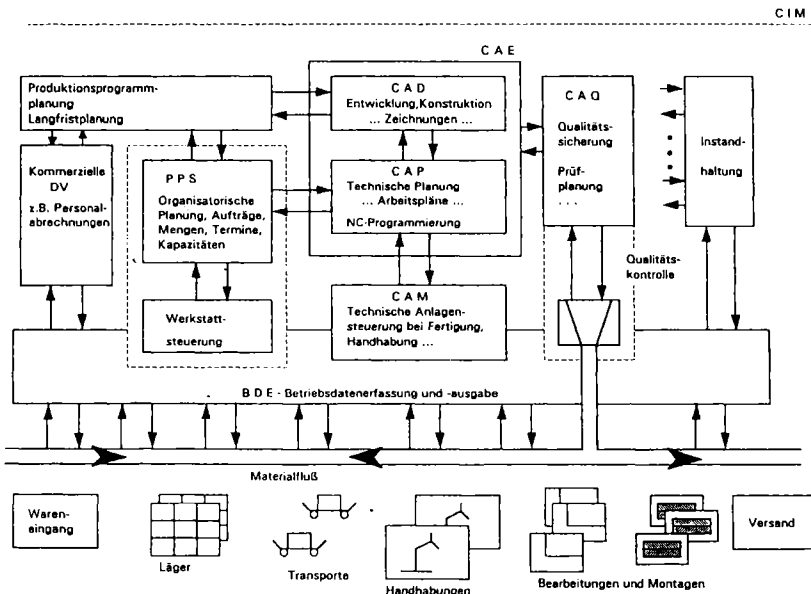


Methoden

B.-Lösungen haben ihre Berechtigung von klein bis ganz groß (z.B. nach der Zahl der *B.-Terminals* gerechnet), von einfach bis komplex schon von den Aufgaben her, von eigenständig bis integriert (*CIM/B.-Konzept*).

B. hat im Rahmen von CIM der rechnergestützten Produktion eine wesentliche Dienstleistungsaufgabe. Bei CIM sind die verschiedensten DV-Anwendungen integriert zu realisieren. Dies liefert z.B. eine sonst nicht mögliche Schnelligkeit der Aufgabenbewältigung. *B.* muß dabei ein ständig aktuelles Abbild sämtlicher betrieblicher Teilbereiche datenmäßig liefern und alle Datenerfassungen und -aufgaben erledigen, die andere CIM-Komponenten nicht erbringen. Werden in der Abbildung die CIM-Komponenten ↑PPS als organisatorische, ↑CAD/↑CAM sowie ↑CAQ u.a. als technische Integrationsblöcke betrachtet, streben diese alle vertikale Datendurchgängigkeiten an; die *B.* ergänzt sie auf der betrieblichen Ebene horizontal (Roschmann, 1991), so wie dies eine geeignete Datenbanklösung auf der EDV-Seite tut.

Literatur: Baron et al.: Mensch - Computer - Information. Akzeptanz und Wirtschaftlichkeit EDV-gestützter Betriebsdatenerfassungssysteme, Eschborn/Ts., 1987. Müller, W., Störmer, W.: Personalzeit- und Betriebsdatenerfassung,



Neuwied, 1993. Roschmann, K., Jung-hanns, J.: Zeit- und Betriebsdatenerfassung. Die Bibliothek der Technik, Bd. 81, Landsberg/L., 1993. Roschmann, K.: Betriebsdatenerfassung. in: Geitner, U.W. (Hrsg.), CIM-Handbuch, 2. Aufl., Braunschweig, Wiesbaden, 1991. Roschmann, K.: CIM-Kette Produktionsplanung und -steuerung - PPS, in: Schüler, U. (Hrsg.), CIM-Lehrbuch - Grundlagen der rechnerintegrierten Produktion, Braunschweig, 1994. Virnich, M., Posten, K.: Handbuch der codierten Datenträger - Identifikationssysteme für Produktion, Logistik, Handel und Dienstleistung, Köln, 1992.

Prof. Dr. K. Roschmann, Konstanz

Benutzerfreundlichkeit

Subjektiver Qualitätsmaßstab über den Aufwand, den der Benutzer leisten muß, um ein DV-System anzuwenden.

Benutzungsoberfläche

Systemkomponenten, über die der Benutzer mit den eigentlichen, problemorientierten Teilen des Systems kommunizieren; heute besitzen die meisten Softwareprodukte graphische Benutzungsoberflächen, die darüber hinaus das Zeigen und Zeichnen mit einer Maus und die Darstellung detailreicher Zeichnungen und Bilder unterstützen; zukünftige, multimediale Nutzungsoberflächen umfassen auch akustische und haptische Ein- und Ausgabe, sowie Bewegtbilder bis hin zu künstlichen, virtuellen Welten.

Benutzungsoberflächen, graphische

Begriffserklärung

Unter der *Benutzungsoberfläche* eines Programmsystems werden diejenigen Systemkomponenten verstanden, über die der Benutzer mit den eigentlichen, problemorientierten Teilen des Systems kommuniziert. Die früher üblichen alphanumerischen Benutzungsoberflächen ermöglichten lediglich die Eingabe von Zeichen über eine Tastatur und die Ausgabe von Zeichen auf einen alphanumerischen Drucker oder Bildschirm. Heute besitzen die meisten Softwareprodukte *B.*, die darüber hinaus das Zeigen und Zeichnen mit einer Maus und die Darstellung detailreicher Zeichnungen und Bilder unterstützen. Künftige, multimediale Benutzungsoberflächen umfas-

sen auch akustische und haptische Ein-/Ausgabe, sowie Bewegtbilder bis hin zu künstlichen, virtuellen Welten.

Eine *B.* stellt sich heute dem Benutzer als Menge von Fenstern, Menüs, Bedienungsfeldern und bildhaften Symbolen dar, die er vornehmlich durch Zeigen und Verschieben mit der Maus, dem Trackball, teilweise sogar mit dem Finger, manipuliert, während die Tastatur praktisch nur noch der Eingabe von Textdaten und weniger häufig benötigter Befehle dient. Gegenstände und Vorgänge des Alltags werden metaphorisch nachgeahmt - so wird ein Objekt in den Papierkorb gelegt, statt es mittels eines Befehls zu löschen; dadurch wird die Benutzung komplexer Programmsysteme einfacher und weniger fehleranfällig.

Die Entwicklung *B.* hat insgesamt zu einem tiefgreifenden Paradigmenwechsel in der Benutzung von Programmsystemen geführt. Aufgrund des beschränkten Eingaberepertoires mußte sich der Benutzer eines alphanumerisch orientierten Systems zumeist merken, mit welchem Programm er gerade kommunizierte und in welchem Modus sich dieses befand. Charakteristisch hierfür sind alphanumerische Textsysteme mit Einfüge-, Überschreib- und Kommandomodus. Die reicheren Darstellungsmöglichkeiten *B.* machen solche Modi überflüssig oder zumindest unmißverständlich sichtbar; in gut entworfenen modernen Fenstersystemen kann sogar mit mehreren Programmen gleichzeitig gearbeitet werden (externe im Gegensatz zu interner Kontrolle); es ist der Benutzer, der das System steuert und nicht umgekehrt.

Architektur

Der Ablauf eines interaktiven Programms ist am besten als Rückkopplungsschleife zu verstehen. Zur Verdeutlichung wird als Beispiel ein Programm zum Entwurf elektronischer Schaltungen herangezogen. Der Benutzer gibt Daten (Punkte des Schaltplans) und Befehle ("verbinde") ein und beeinflusst so den Systemzustand, der anschließend in geeigneter Form (Schaltplan mit neuen Verbindungen) dargestellt wird. Das Beispiel demonstriert die Aufgaben, die dabei gelöst werden müssen:

- Identifikation dessen, was eingege-

ben wurde, mit internen Daten bzw. Befehlen: So wird ein ungenau gezeigter Bildschirmpunkt mit dem nächstliegenden Verbindungspunkt des internen Schaltplanmodells identifiziert.

- Prüfung der Eingabe auf Zulässigkeit: Im Beispiel ist etwa die Verbindung eines Punktes mit sich selbst sinnlos.

- Verarbeitung der Eingabe und Transformation des Systemzustands: Hier werden z.B. die elektronischen Eigenschaften der geänderten Schaltung geprüft und ein neues, kreuzungsfreies Layout der Leitbahnen berechnet.

- Abbildung des internen Systemzustands auf das Ausgabemedium: Das Schaltplanmodell wird mit den für das Verständnis erforderlichen Details (z.B. bildhafte Darstellung von Elementarschaltkreisen) graphisch dargestellt.

Nur die Aufgabe 3 fällt dem eigentlichen Anwendungsprogramm zu. Die Aufgaben 1, 2 und 4 werden durch die graphische Benutzungsoberfläche erledigt. Entsprechend dem Seeheim-Referenzmodell (Pfaff, 1985) zerfällt sie daher in folgende Komponenten:

- Das *Anwendungssystem* verwaltet die problemorientierten *Datenstrukturen*, ebenso führt es die eigentlichen Berechnungen durch.

- Die *Anwendungsschnittstelle* bildet das Bindeglied zwischen Anwendung und Benutzungsoberfläche.

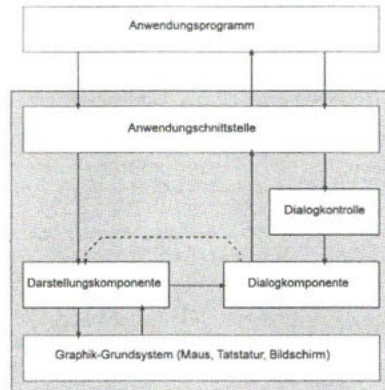
- Die *Darstellungskomponente* abstrahiert von konkreten durch das graphische Grundsystem bestimmten Gegebenheiten, wie Schrift, Auflösung, absolute Mauskoordination usw.; sie setzt primitive Ereignisse (Mausbewegungen in höhere, wie das Betreten oder Verlassen eines Fensterbereichs) um und kann z.B. auch dafür verantwortlich sein, Fensterinhalte, die inkonsistent geworden sind, durch Neuzeichnen aufzufrischen.

- Mit Hilfe der *Dialogkontrolle* kann das Anwendungsprogramm bestimmen, welche Dialogfunktionen dem Benutzer zur Verfügung stehen, wie Reihenfolge-, Vollständigkeits- und Konsistenzkriterien spezifiziert und festlegt, auf welchen anwendungsorientierten Ereignisse die Benutzeraktionen abgebildet werden.

- Die *Dialogkomponente* hat drei Aufgaben. Zunächst entscheidet sie anhand

der Dialogkontrollinformation, ob ein von der Darstellungskomponente erkanntes Ereignis überhaupt an die Anwendung weitergereicht wird oder nicht. Zweitens führt sie in Rückkopplung mit der Darstellungskomponente "direkte Manipulationen", wie die Auswahl aus einem aufrollenden Menü oder das Verstellen eines Rollbalkens, durch und gibt lediglich das Resultat an die Anwendung weiter. Dies ist in der Abbildung durch die gestrichelte Schleife symbolisiert. Schließlich steuert die Dialogkomponente bei interner Kontrolle den Dialog - gemäß dem augenblicklichen Zustand des Anwendungsprogramms.

- Das *graphische Grundsystem* führt die eigentlichen Zeichenoperationen durch, es entdeckt Ereignisse, wie einen Tastenanschlag oder einen Mausklick und leitet sie an die Darstellungskomponente weiter.



Wichtigstes Ziel dieser Zerlegung ist eine starke Entkopplung zwischen Anwendung und Benutzungsoberfläche. Die gleiche Oberflächensoftware (†User Interface Management System) kann so in einer Vielzahl von Anwendungsprogrammen genutzt werden; sie senkt nicht nur den Programmieraufwand, sondern führt zu einer wünschenswerten Vereinheitlichung (einheitliches Touch and Feel). Wirklich erreicht wird eine solche Vereinheitlichung jedoch erst, wenn die graphisch-interaktive Repräsentation semantisch gleichartiger Objekte und Vorgänge durch zusätzliche Richtlinien (Style Guide) erzwungen wird.

Interaktionsobjekte

Aus Sicht des Benutzers besteht eine *B.* aus Interaktionsobjekten, die er mittels Mausclick und Tastatur manipuliert. Dabei werden einfache Interaktionsobjekte zu komplexeren (sog. Widgets) zusammengesetzt:

- *Dokumentfenster* dienen der Eingabe größerer Datenmengen; sie können hierzu - gleich einem Formular in Einzelbereiche untergliedert sein. Fenster sind gewöhnlich mit weiteren Interaktionsobjekten, z.B. "Knöpfen" zum Schließen, "Griffen zum Verkleinern und Vergrößern" oder "Schiebern" zum Verschieben des Fensterinhalts ausgestattet.

- *Dialogfenster* sind gewöhnlich nur zeitweise sichtbar; sie dienen der Eingabe von Kommandos, der Einstellung von Parametern und der Übermittlung von *↑*Nachrichten, Warnungen oder Fehlern, die der Benutzer durch Knopfdruck quittieren muß.

- *↑Menüs* dienen der Auswahl von Kommandos oder *↑*Daten. Sie sind meist hierarchisch organisiert und mit den Interaktionsobjekten, auf die sie wirken sollen, verknüpft.

- *Knöpfe, Schieber, Rollisten* und dgl. sind meist Fenstern zugeordnet und besitzen von diesen abhängige Funktionen.

- *Bildsymbole* (Icons) repräsentieren aus Platzgründen verkleinerte oder zeitweilig beiseite gelegte Daten, die zu Fenstern expandiert (geöffnet) oder zwischen Fenstern bewegt (kopiert) werden. Wichtigstes Eingabegerät ist die *Maus*, die als variabel geformter Zeiger (*↑Cursor*) auf dem Bildschirm erscheint. Die Form des Zeigers soll den Benutzer über die Wirkung informieren, die ein Druck auf einen der Mausknöpfe an der gezeigten Stelle hätte. Da der Benutzer den Mauszeiger fast ständig im Blickfeld hat und verfolgt, können ihm so auch weitere Mitteilungen gemacht werden - z.B. weist ein Uhrensymbol darauf hin, daß die vorangegangene Aktion noch nicht abgeschlossen ist.

Produkte

Im Bereich kleiner und mittlerer Rechner (*↑*Betriebssysteme MS-DOS, UNIX) haben sich eine Reihe von verschiedenen Standardprodukten für *B.* etabliert. MS-Windows und Windows NT können fast als eigenständige "graphische Betriebs-

steme" angesehen werden, während sich in der UNIX-Welt aus X-Windows der herstellerübergreifende Standard *↑*OSF Motif entwickelt hat, der vielen herstellereigenen *↑*Benutzungsoberflächen zugrunde liegt.

Literatur: Pfaff, G.E. (Hrsg.): User Interface Management Systems, 1985.

Prof. Dr. J. Röhrich, Jena

Beratung, wissensbasierte Systeme in

*↑*wissensbasierte Systeme in der Beratung

Berufsfelder der Wirtschaftsinformatik

Die *Wirtschaftsinformatik* ist die Wissenschaft von dem Entwurf und von der Anwendung computergestützter *↑*Informationssysteme. Sie betrachtet alle Fragen der Gestaltung von *↑*Anwendungssystemen in Wirtschaft und Verwaltung. Die Ausbildung im Fachgebiet Wirtschaftsinformatik ist mehrfach gegliedert. Die Empfehlung der *↑*GI unterscheidet:

- Informatik-Studium mit Schwerpunkt fach Wirtschaftsinformatik,
- vollzügiger Studiengang Wirtschaftsinformatik,
- Betriebswirtschaftliches Studium mit Schwerpunkt Wirtschaftsinformatik,
- Betriebswirtschaftliches Studium mit Schwerpunkt Wirtschaftsinformatik.

Gemäß dieser unterschiedlichen Schwerpunktbildung ist auch das Tätigkeitsfeld mehr oder weniger stark im Spannungsfeld zwischen Informatik und Wirtschaftswissenschaft ausgerichtet.

Tätigkeitsmerkmale sind die folgenden:

- Der *wissenschaftlich tätige Wirtschaftsinformatiker* hat die Möglichkeit, den Schwerpunkt der zu bearbeitenden Probleme an persönlichen Interessenschwerpunkten auszurichten. Er ist anwendungsorientierter Forscher und Dozent bzw. Ausbilder. Die Aufgabenbewältigung steht weniger unter dem Druck von Wirtschaftlichkeitserfordernissen.

- Der *Wirtschaftsinformatiker in Wirtschaft und Verwaltung* hat häufig eine Vermittlerfunktion. Er hat die Aufgabe, zwischen den Anforderungen der Mitarbeiter und Nutzer der Fachabteilung auf der einen Seite und Systementwicklern und Betreibern auf der anderen Seite einen Interessenausgleich zu erreichen. Organisatorisch ist er meist einer dieser Gruppen zugeordnet, seltener betreut er solche Aufga-

ben aus einer über- oder nebengeordneten Stabsstelle.

Die Arbeit an der Weiterentwicklung und Implementierung von ↑Anwendungssystemen erfolgt in Teams zusammen mit Spezialisten der DV-Abteilung und der Fachabteilung. Der Wirtschaftsinformatiker muß die Aufgabenbereiche des ↑Managements kennen und beherrschen, daneben aber auch die technischen Bereiche überschauen, damit er die Fachsprache der DV-Spezialisten versteht. Die Arbeit vollzieht sich meist in Projekten, die einen zeitlich, inhaltlich und budgetmäßig festen Rahmen haben. Die Projektarbeit erfordert einen großen Fundus von Methoden- und Anwendungswissen und auch die Bereitschaft, neue Methoden und Techniken aufzugreifen und einzuführen.

Der wachsende Markt, das Spektrum der ↑Anwendungssysteme, sowie deren Leistungsumfang erfordern große Verantwortung bei der Auswahl von Anwendungssystemen. Die Neu- und Eigenentwicklung von Anwendungssystemen tritt mehr und mehr in den Hintergrund und ist eher noch bei firmenspezifischen Problemen erforderlich. Die weitgehende Durchdringung der Betriebe mit ↑Informationssystemen erfordert bei zunehmender Integration der ↑Systeme aber auch bei den nicht selten vorkommenden Zusammenführungen verschiedener Betriebe Kenntnisse von Migrationsstrategien. Angestrebte Vereinfachungen können zu geeigneten Auslagerungen von Teilfunktionen führen. Diese müssen in technischer und/oder in finanzieller Hinsicht Vorteile bringen. Die Tätigkeiten erfordern, daß der Wirtschaftsinformatiker durch Weiterbildung über die Fachliteratur, aber auch Seminare und Tagungen Kenntnisse über die neuesten Entwicklungen bekommt.

Die Berufe in der ↑Datenverarbeitung sind einem ständigen Wandel unterworfen. Einen guten Überblick liefern die Blätter zur Berufskunde, die bei den Arbeitsämtern erhältlich sind. Die *Berufsfelder im einzelnen* sind die folgenden:

- *Tätigkeiten im DV-Bereich des Betriebs (Führungsposition):* Der Berufsanfänger wird in einer problemorientierten Anwendergruppe eingesetzt. Die Aufgabenschwerpunkte sind Gestaltung und Verbesserung von Büro-, Kommunikations- und Anwendungssystemen in den betrieb-

lichen Funktionsbereichen. Nach erfolgreicher Bearbeitung einer oder mehrerer Projekte erfolgt eine Tätigkeit als Gruppen-, Projekt- und später als Organisationsleiter. Das Aufgabenspektrum wandelt sich in den letzten Jahren mehr und mehr zu strategischen Bereichen (↑Informationsmanagement). Darum wird die DV-Leitung zu einer Managementfunktion für die ↑Informationsverarbeitung und Kommunikation.

- *Tätigkeiten im DV-Bereich des Betriebs (Benutzer):* Das Aufgabenfeld hat einen starken unternehmensinternen Bezug. Die Betätigungsfelder sind Erstellung, Wartung von Anwendersoftware, Implementierung, Installation, Anpassung von Softwaresystemen, Konfigurationsgestaltung und Organisationsgestaltung.

- *Tätigkeiten in öffentlichen Betrieben (Verwaltung, Behörde, Ministerium, Hochschule):* Die Aufgaben sind ähnlich und ebenfalls durch eine große Aufgabebreite gekennzeichnet. Typisch ist jedoch, daß die Aufgabenbereiche relativ starr und unflexibel definiert sind und dem Stand der Technik oft nur mit großer Verzögerung angepaßt werden.

- *Tätigkeiten in der Fachabteilung:* Der Wirtschaftsinformatiker ist *Mitglied der Fachabteilung*. Er vertritt die Interessen der Anwenderabteilung bei der Softwareentwicklung und Beschaffung. Er sorgt für einen sachgemäßen Einsatz, paßt die ↑Software an die Erfordernisse an und schult die Anwender. Er benötigt dazu Kenntnisse über die Sachprobleme der eigenen Abteilung, ihrer Ablauf- und Aufbauorganisation aber auch über die ↑Schnittstellen zu anderen betrieblichen Aufgabengebieten.

Für die Aufgabenerfüllung in den Fachbereichen werden die notwendigen DV-Kenntnisse vorausgesetzt, damit er die Mitarbeiter der Fachabteilung sachkundig bei der Gestaltung des DV-Einsatzes unterstützen kann. Der Schwerpunkt liegt jedoch auf den fachbezogenen Tätigkeiten. Als Sachbearbeiter wickelt er die DV-Anwendungen kompetent und sachkundig ab. Als Mittler ist er in der Lage, die Wünsche der Fachabteilung in bezug auf die informationstechnologische Infrastruktur entsprechend zu formulieren.

Tätigkeiten bei einem Hard- und Softwareproduzenten: Als Wirtschaftsinformatiker stehen hier eine Vielzahl von Betätigungsfeldern zur Verfügung, die

- Entwicklung von Anwendungssoftwaresystemen,
- Hilfe bei der Softwareentwicklung
- Produktschulung.

Im Bereich der *Produktentwicklung* unterstützt er die verschiedenen Phasen des Software-Lebenszyklus, beginnend bei der Istanalyse, über den Entwurf, die Realisierung, den Test bis zur Markteinführung. Er kennt Vor- und Nachteile verschiedener Software-Entwicklungswerkzeuge und unterstützt deren Einsatz. Er untersucht am Markt die Anforderungen an sein Produkt und entwickelt ein kundenorientiertes Marketingkonzept. Es schließen sich nach mehrjähriger Tätigkeit in diesem Bereich weitere anspruchsvolle Aufgaben an, als Produktmanager, in der Ausarbeitung von DV-Gesamtplänen, in Managementaufgaben im Bereich der Softwareerstellung, in der Abnahme und Freigabe von Software. Der *Produktmanager* befindet sich an der Schnittstelle zwischen Produktentwicklung und Vertrieb. In dieser Position fallen folgende Tätigkeiten an: Beratung und Erprobung neuer Produkte, Marktbeobachtung und Kundenbefragungen über des eigene Produkt, Konkurrenzprodukte und benachbarte Anwendungsgebiete, Auswahl und Beratung von Softwareprodukten, konzeptionelle Weiterentwicklung des Produktspektrums.

Die Vielfalt der Produkte und deren Komplexität erfordern ein genaue Marktkenntnis und eine gute Produktkenntnis. Die Kundenunterstützung ist sehr wichtig, Softwarefehler müssen registriert und nach geeigneten Verfahren abgewickelt werden. Häufig muß auch eine Anwenderschulung durchgeführt werden. Diese erfordert die Erstellung geeigneter Unterrichtsmaterialien und Handbüchern. Es muß dafür gesorgt werden, daß geeignete Schulungsleiter herangebildet werden.

Viele Firmen, die nicht die erforderliche Größe haben oder denen das geeignete Fachpersonal fehlt, greifen auf kompetente *Unternehmensberater* zurück. Diese können Mitarbeiter in großen Unternehmensberatungsgesellschaften sein, aber auch als Freiberufler ihren Sachverstand anbieten. Er unterstützt und berät bei der Erstellung und Einführung von ↑Anwendungssoftware, der Entwicklung von Rahmenkonzepten, der organisatorischen und DV-

technischen Einführung von Anwendungssystemen.

Als Mitglied großer Beratungsgesellschaften hat er oft die Aufgabe, für die ordnungsgemäße, sichere und funktionsfähige Abwicklung von DV-Aufgaben zu sorgen.

Hier arbeitet er im Team mit Wirtschaftsprüfern. Diese Tätigkeit kann sowohl innerbetrieblich als Funktion der internen Revision als auch im Rahmen der externen Prüfung von einer Wirtschaftsprüfungsgesellschaft durchgeführt werden. Dieser Tätigkeitsbereich umfaßt die Aufgaben Erarbeiten interner Anforderungen und Grundsätzen ordnungsgemäßer DV-Abwicklung, Beratung der Abteilungen, Prüfen von Anwendersoftware einschließlich vor- und nachgelagerter Arbeitsabläufe.

Tätigkeiten in der Lehre und Forschung: Der große Bedarf an Ausbildungseinrichtungen für DV-Anwender bedingt ein breites Angebot an Aus- und Weiterbildung. Diese Ausbildungen werden von Schulen und Hochschulen, privaten und öffentlichen Institutionen, von Anwendern und Herstellern getragen. Die Tätigkeit als *Lehrer* verlangt eine über das normale Maß hinausgehende Beschäftigung mit den neuesten Techniken und Entwicklungen des DV-Marktes. *Forschung* auf dem Gebiet der Wirtschaftsinformatik wird sowohl in der Industrie, in öffentlichen (Groß-) Forschungseinrichtungen und an den Hochschulen betrieben. Die genannten Institutionen pflegen häufig eine intensive Zusammenarbeit. Ein beruflicher Wechsel zwischen den einzelnen Institutionen ist möglich.

Für praxisorientierte Wirtschaftsinformatiker bzw. Diplomkaufleute mit entsprechender Vertiefung besteht bei wissenschaftlicher Qualifikation und mehrjähriger Berufspraxis die Möglichkeit, Professor/ Professorin einer Fachhochschule zu werden.

Literatur: GI-Arbeitskreis 7.1.2 (Informatik an Fachhochschulen): Neue Empfehlungen der Gesellschaft für Informatik für das Informatikstudium an Fachhochschulen. Informatik Spektrum, Bd. 7, H. 3, 1984. GI-Arbeitskreis 7.1.2 (Wirtschaftsinformatik in wirtschaftswissenschaftlichen Studiengängen an Fachhochschulen) im Fachausschuß 7.1: Empfehlungen zur Integration der Wirtschaftsinformatik- Informatik Spektrum, Bd. 13, H. 3, 1984. Diplom-

Informatiker, Blätter zur Berufskunde, Best.Nr. 3-I A 02. Diplom-Kaufmann, Blätter zur Berufskunde, Best.Nr. 3- IX A 04). Diplom-Informatiker (FH), (Blätter zur Berufskunde, Best.Nr. 2- I A 31). Diplom-Wirtschaftsinformatiker(in) (Fachhochschule), Diplom-Informatiker (in) der Fachrichtung Wirtschaftsinformatik, Diplom-Betriebswirt(in) (Fachhochschule) mit Schwerpunkt Wirtschaftsinformatik, Blätter zur Berufskunde, Best.Nr. 3-I A 44). Wirtschaftsinformatiker/Wirtschaftsinformatikerin(BA), Blätter zur Berufskunde, Best.Nr. 2- IX A 46. Bundesanstalt für Arbeit (Hrsg.): Berufe in der Datenverarbeitung. Blätter zur Berufskunde Bd. O-2200, Nürnberg, 1990.

Prof. Dr. H. Gipper, Aachen

betriebliche Anwendungen,

Optimierungssystem für

↑Optimierungssystem für betriebliche Anwendungen

betriebliche Umweltinformationssysteme

↑Umweltinformationssysteme, betriebliche

betriebliche Umweltinformationssysteme und Umwelt-PPs

↑Umweltinformationssysteme und Umwelt-PPs, betriebliche

Betriebsart

Mode of Operation; Art und Weise der Aufgabenabwicklung.

Betriebsdaten

Die im Laufe eines betrieblichen Prozesses anfallenden bzw. verwendeten Daten; es handelt sich um technische und organisatorische Daten über das Verhalten bzw. über den Zustand des Betriebes, so bspw. Angaben über produzierte Mengen, benötigte Zeiten, Zustände von Fertigungsanlagen, Lagerbewegungen, Qualitätsmerkmale.

Betriebsmodelle

Grundlagen

B. beschreiben Aufbau und Funktionsweise derjenigen Systeme eines Unternehmens, die allgemein der Leistungserstellung und insb. der Produktion dienen. Als Konstrukte der betriebswirtschaftlichen Theorie modellieren sie somit Organisation und Technologie von Produktionssystemen i.w.S. *B.* sind das Ergebnis

eines Abstraktionsprozesses. Dieser ist generalisierend, da von allen Gegebenheiten der Produktion bis auf das ökonomisch allgemeingültige abgesehen wird, isolierend, insoweit das quantitative und zeitliche Moment des Produktionsgeschehens aus dem umfassenden technischen und gesellschaftlichen Zusammenhang herausgelöst und verselbständigt wird und idealisierend, weil der Produktionsprozeß, der sich in Wirklichkeit im Zeitkontinuum analog vollzieht, im Modell durch ein Mengen- und Zeitgerüst künstlich diskretisiert und somit nur angenähert dargestellt wird.

Die \uparrow Objekte, die den *B.* zugrunde liegen, lassen sich durch Mengen- und Zeitgrößen oder als monetäre Größen darstellen. Insofern bilden *B.* entweder Produktionsmodelle oder Preis- und Kostenmodelle. Die Modellobjekte beziehen sich letztlich auf Güter in ihren Rollen als Produktionsobjekte und Produktionsmittel, die in der Produktionstechnik grundsätzlich in Form von Einsatz (Input) oder Ausbringung (Output) verwendet werden.

Die technologischen Gesetzmäßigkeiten, nach denen Input und Output erfolgen, werden bei *B.* durch Aktivitätenfunktionen modelliert. Eine Aktivitätenfunktion ist eine vektorwertige Abbildung $r: X \rightarrow R_+^n$, die jedem Argument $x \in X \subseteq R_+^n$ dasjenige Bild $r(x)$ zuordnet, das mit ersterem technologisch zur Aktivität kombinierbar ist. Insofern läßt sich der Graph einer Aktivitätenfunktion als Aktivitätsmenge verstehen, die dem Produzenten zufolge seiner Technologie zur Wahl steht. Die Aktivitätenfunktion läßt zwei Interpretationen zu. Unter der Inputsicht wird das Argument $x \in X$ als Output und sein Bild unter der Funktion r als Input gedeutet. Die Aktivitätenfunktion heißt in diesem Fall Inputfunktion. Die Umkehrung dieser Sicht führt zur Interpretation der Aktivitätenfunktion im Sinne einer Outputfunktion. Die Aktivitätenfunktion kann sogar komponentenweise abwechselnd nach Input- oder Outputsicht interpretiert werden, wenn die Modellierung der Technologie dies erfordert. Die einander entsprechenden Komponenten des Arguments und seines Bildes unter einer Aktivitätenfunktion beziehen sich jeweils

auf dasselbe Modellobjekt, so daß die Differenz aus beiden stets sinnvoll definiert und als Ergebnis der Aktivität zu deuten ist. Unter der Inputsicht ist die Differenz $x-r(x)$ für $x \in X$ komponentenweise als Überschuß der Ausbringung eines Modellobjektes über seinen Wiedereinsatz oder als Unterschuß (Fehlbetrag) zu deuten. Im Fall der Outputsicht sind die Ergebniskomponenten bei Umkehrung des Vorzeichens entsprechend zu interpretieren. Die Ergebnisfunktion ergibt sich dann als $t := id_x r$ zu r und erzeugt als Bildmenge $t[X]$ des Definitionsbereichs X die Technologiemenge des Produzenten im Sinne der Aktivitätsanalyse. Während eine Aktivität $(x, r(x))$ für $x \in X$ das Innenverhalten des Produktionssystems beschreibt, wird durch ihr Ergebnis $t(x)$ das Außenverhalten dargestellt. Falls positive oder negative Ergebniskomponenten Güterquantitäten darstellen, bedeuten sie bei der Inputsicht Produktquantitäten, die das Produktionssystem an sein Umsystem abgibt bzw. Faktorquantitäten, die es von dort empfängt; in der Outputsicht ist die Interpretation der Ergebniskomponenten bei Umkehrung des Vorzeichens entsprechend. Die Produktionsaufgabe, die ein Produzent mit Hilfe seiner Technologie zu erfüllen hat, setzt sich aus einem Sachziel und einem Formalziel zusammen. Das Sachziel besteht darin, im Ergebnis einer Aktivität eine vorgegebene untere Schranke einzuhalten. Bei der Inputsicht enthält die Schranke positive Vorgaben für Produktquantitäten, die mindestens ausgebracht und an das Umsystem abgeführt werden müssen, und negative Vorgaben für Faktorquantitäten, die betragsmäßig höchstens aus dem Umsystem dem Produktionssystem zugeführt und dort verbraucht werden dürfen. Unter der Outputsicht sind die Vorgabenkomponenten bei Umkehrung des Vorzeichens entsprechend zu deuten. Wenn für eine Planungsvorgabe mehrere Aktivitäten existieren, die im Ergebnis das Sachziel erfüllen, muß der Produzent bzgl. der alternativen Aktivitäten eine Auswahlvorschrift, d.h. das Formalziel festlegen. In diesem Sinne ergibt sich die Planungsaufgabe durch die Planungsungleichung $y \leq t(x)$, worin y die Planungsvorgabe bedeutet und der auf ihrer Lösungsmenge

definierten Auswahlvorschrift (Steffens, 1987). Für zwei Vektoren v, v' gelte $v \leq v'$ genau dann, wenn die \leq -Beziehung komponentenweise erfüllt ist.

Das Produktionssystem ist das Original, von dem der Produzent ein \uparrow Betriebsmodell bildet. Dabei kann es selbst wieder aus Produktionssystemen bestehen. Der Aufbau eines Produktionssystems und somit auch seines Modells läßt sich auf kleinste Bausteine zurückführen. Diese entstehen durch technologische Verfahren, die mit Hilfe von Produktionsmitteln an Produktionsobjekten vollzogen werden. Die Anwendung eines solchen Verfahrens läßt sich durch eine Aktivitätenfunktion beschreiben, die als Modell eines atomaren Produktionssystems verstanden werden kann. Jede Aktivität, die die Funktion erfüllt, modelliert eine mögliche Realisierung des Verfahrens. Die Verfahren, über die ein Produzent insgesamt verfügt, bilden i.a. einen Zusammenhang, weil Produktionsobjekte von einem Verfahren ausgebracht und in einem anderen wieder eingesetzt werden.

Außerdem können zu einem Produktionsobjekt alternative Verfahren existieren, durch die das \uparrow Objekt entweder ausgebracht oder eingesetzt wird. Eine Menge solcher Verfahren bildet zusammen mit den durch sie bearbeiteten Produktionsobjekten und beanspruchten Produktionsmitteln ein Produktionssystem. Soll hiervon ein Betriebsmodell aufgestellt werden, so müssen zuvor unter der Inputsicht alle Wahlprobleme zwischen alternativen Ausbringungsverfahren, sowie unter der Outputsicht alle Wahlprobleme zwischen alternativen Einsatzverfahren gelöst werden. Ist dies erfolgt, so gelten die verbleibenden Aktivitätenfunktionen, die Modelle von Verfahren sind, im \uparrow Betriebsmodell simultan und nicht mehr alternativ. Diese können nun zur Aktivitätenfunktion des umfassenden Produktionssystems aggregiert werden. Dabei wird für zwei beliebige verfahrensbezogene Aktivitätenfunktionen vorausgesetzt, daß ihnen dieselbe Menge von Modellobjekten zugrunde liegt und sich die einander entsprechenden Komponenten ihrer Definitions- bzw. Wertebereiche stets auf dasselbe Modellobjekt beziehen.

Lineare Modelle

B. gehen auf *Input/Output-Modelle* zurück, die von Leontief für die Analyse gesamtwirtschaftlicher Zusammenhänge eingeführt worden sind (Leontief, 1966). Die volkswirtschaftliche *Input/Output-Analyse* umfaßt im wesentlichen zwei Hauptgebiete: die statische und die dynamische Theorie (Schumann, 1968). Dem statischen, offenen *Input/Output-System* kommt dabei eine besondere Bedeutung zu. Die Aktivitätsfunktion dieses Systems wird als lineare Inputfunktion auf dem gesamten \mathbb{R}^n definiert; sie läßt sich stets durch eine $(n \times n)$ -Matrix R , die komponentenweise nicht negativ und auf beliebige Outputvektoren $x \in \mathbb{R}^n$ anzuwenden ist, darstellen. Jede Spalte dieser Matrix entspricht genau einem volkswirtschaftlichen Sektor, der genau ein Gut ausbringen kann, so daß sich die Inputfunktion des Sektors als Anwendung seines Spaltenvektors auf die entsprechende Komponente des Outputvektors ergibt. Da der Definitionsbereich dieser Inputfunktion nicht weiter eingeschränkt wird, lassen sich auch keine technologisch bedingten Restriktionen, wie z.B. die der Kuppelproduktion darstellen. Darüber hinaus wird das Modell durch die sogenannte *Workability-Annahme* eingeschränkt. Sie besagt, daß zu der Technologiematrix $(E-R)$ stets eine Inverse existiert, die komponentenweise nicht negativ ist. Dadurch wird letztlich die eindeutige Lösbarkeit der Planungsaufgabe garantiert und die Lösung durch Anwendung von $(E-R)^{-1}$ auf eine Planungsvorgabe y gewonnen. Für diese *Workability-Bedingung* sind mehrere äquivalente Umformulierungen nachgewiesen worden. Eine Besonderheit des Leontief *Input/Output-Modells* besteht darin, daß zu ihm, dem primalen Modell, ein duales existiert, das als Preismodell zu deuten ist (Nikaido, 1968).

Die von Leontief begründete *Input/Output-Theorie* wurde bereits in den 50er Jahren auf betriebswirtschaftliche Probleme der Produktionsplanung und Kostenüberwachung angewandt. Dabei sind vor allem die *B.* von Pichler und Wenke bekannt geworden. Das Pichler-Modell, das breite Beachtung fand, hat ein Industriekraftwerk zum Gegenstand (Pichler, 1959). Die Untersuchungen von Wenke

beziehen sich auf die Verbundwirtschaft in der Großchemie (Wenke, 1959). Diese *B.* waren in den beiden folgenden Jahrzehnten Vorbilder für die Entwicklung und Anwendung weiterer *B.* vor allem in der eisen- und stahlerzeugenden Industrie (Franke, 1972). Die Betriebswirtschaftslehre hat sich in der Folgezeit zur Aufgabe gemacht, die linearen *B.* zu verallgemeinern und auf eine breitere theoretische Grundlage zu stellen. In diesem Zusammenhang wurden auch nichtlineare (Kloock, 1969) und dynamische (Küpper, 1979) *Input/Output-Systeme* aufgestellt und untersucht. Alle diese Modelle basieren letztlich auf der Struktur der Leontief *Input/Output-Modelle*, bei denen davon ausgegangen wird, daß die Ausbringungen der Sektoren voneinander unabhängig sind. Diese Annahme ist jedoch für zahlreiche betriebswirtschaftliche Anwendungen von *Input/Output-Modellen* zu restriktiv.

Isotone B. (Steffens, 1987) zeichnen sich im wesentlichen durch zwei Forderungen aus: Erstens wird vom Definitionsbereich $X \subseteq \mathbb{R}^n$ der Aktivitätenfunktion r verlangt, daß er infimumvollständig ist, d.h. das Infimum jeder nichtleeren Teilmenge von X selbst Element von X ist. Zweitens wird von der Aktivitätenfunktion verlangt, daß sie isotone ist, d.h. $x \leq x' \Rightarrow r(x) \leq r(x')$. Diese Annahmen wirken empirisch wenig restriktiv und gestatten daher, nahezu alle relevanten Produktionsstrukturen zu modellieren. Insb. erlauben sie jede Kopplung von Komponenten des Argumentvektors, soweit sie die Infimumvollständigkeit des Definitionsbereichs nicht verletzt und somit auch die Behandlung der Kuppelproduktion. Die bisher bekannt gewordenen *B.* lassen sich als isotone Spezialfälle auffassen.

Ein zentraler Satz der Theorie isotoner *B.* besagt, daß bereits aus der Isotonie der Aktivitätenfunktion und der Infimumvollständigkeit ihres Definitionsbereichs zu jeder lösbaren Planungsungleichung $y \leq t(x)$ die Existenz einer eindeutig minimalen Lösung $x^* \in X$ folgt. Wegen der Isotonie von r ist auch die Aktivität $(x^*, r(x^*))$ eindeutig minimal unter allen Aktivitäten, die im Ergebnis die Planungsvorgabe y erfüllen. Es bietet sich daher an, die Minimierung der Aktivität zum formalen Auswahlkriterium zu er-

heben. Da aber letztlich nur das Außenverhalten $t(x)$ eines Produzenten und nicht sein Innenverhalten $(x, r(x))$ vom Markt honoriert wird, sind die Eigenschaften der eindeutig minimalen Aktivität $(x^*, r(x^*))$ zu untersuchen. Es läßt sich zeigen, daß diese Aktivität immer zu einem maximalen Ergebnis führt, wenn im Produktionssystem mindestens ein Modellobjekt existiert, dessen entsprechende Ergebniskomponente sich stets echt verkleinert, wenn der Argumentvektor in mindestens einer Komponente echt vergrößert und in den anderen nicht verkleinert wird. Dieser Sachverhalt ist im Anwendungsfall in der Regel gegeben, so daß obige Auswahlvorschrift dann auch ökonomisch sinnvoll ist.

Zur Lösung einer Planungsaufgabe muß der Produzent die eindeutig minimale Aktivität berechnen, die seine Planungsvorgabe im Ergebnis erfüllt. Dazu ist ein Iterationsverfahren bekannt, das Gegenstand eines weiteren zentralen Satzes der Theorie isotoner B . ist. Dieses Verfahren ermittelt die eindeutig minimale Lösung $x^* \in X$ der Planungsungleichung $y \leq t(x)$ als Grenzwert der Folge $(x^v)_{v \in \mathbb{N}}$ mit $x^1 := \min X$ und $x^{v+1} := \inf \{ x \in X \mid x \geq (y + r(x^v)) \vee x^v \}$ für $v \in \mathbb{N}$. Es basiert auf den beiden Zusatzvoraussetzungen, daß jede isotone Folge aus X in X konvergiert und r linksstetig ist. Hierbei handelt es sich um technische Annahmen, die die praktische Anwendbarkeit nicht einschränken. Das angegebene Verfahren erweist sich in der Anwendung als sehr effizient und hat den Vorzug, vor allem auf B . mit zyklischen Produktionsstrukturen anwendbar zu sein. Wird die Planungsvorgabe in mindestens einer Komponente vergrößert, so verkleinert sich die eindeutig minimale Lösung der entsprechenden Planungsungleichung in keiner Komponente. Als Anfangsglied der Iterationsfolge läßt sich jeder Vektor $x \in [\min X, x^*] \cap X$ wählen. Somit ist es möglich, eine laufende Iteration in einem beliebigen Schritt zu unterbrechen und für eine vergrößerte Planungsvorgabe fortzusetzen.

Ein Modellierungsvorteil isotoner B . besteht darin, daß der Definitionsbereich einer Aktivitätsfunktion vom \mathbb{R}_+^n auf solche Teilmengen X eingeschränkt werden kann, die mit der darauf definierten Ak-

tivitätsfunktion genau die Technologie des Produzenten beschreiben und dabei die empirisch wenig restriktive Forderung der Infimumvollständigkeit erfüllen. Jeder infimumvollständige Definitionsbereich X einer Aktivitätsfunktion läßt sich durch eine Kopplungsfunktion $k: X \rightarrow X$ mit $k(\bar{x}) = \inf \{ x \in X \mid x \geq \bar{x} \}$ für $\bar{x} \in X$ $:= \{ x' \in \mathbb{R}_+^n \mid \exists x \in X: \min X \leq x' \leq x \}$ definieren. Somit läßt sich die Iterationsfolge des Verfahrens auch $x^{v+1} := k((y + r(x^v)) \vee x^v)$ für $v \in \mathbb{N}$ schreiben.

Die Semantik einer B . entsteht primär durch Interpretation seiner Modellobjekte. Diese beziehen sich immer auf Güter, die entweder Gegenstand der Produktionsprozesse sind und daher Produktionsobjekte genannt werden, oder die instrumentelle Grundlage bilden, auf der sich die Prozesse vollziehen und daher Produktionsmittel heißen. Die Verwendung der Güter erfolgt formal stets als Input und Output und kann dabei entweder mengen- oder wertmäßig oder zeitlich dargestellt werden. Die dazu eingeführten Größenpaare aus Input und Output, die den Modellobjekten umkehrbar eindeutig entsprechen, beschreiben die Produktionsprozesse im o.g. Sinne künstlich diskretisiert und bilden das Mengen- bzw. Wertegerüst und das Zeitgerüst der B . (Steffens, 1987).

Die Aktivität ist ein \uparrow Tupel aus einem Urbild- und ihrem Bildvektor unter der Aktivitätsfunktion, das als zwispaltige Matrix aufgefaßt werden kann. Jede Zeile dieser Matrix entspricht also genau einem Modellobjekt und die Differenz zwischen der ersten und zweiten Spalte ist gleich dem Ergebnis der Aktivität, das in der Planungsungleichung durch die Vorgabe nach unten begrenzt ist. Dabei ist zunächst noch offen, welche Komponente einer Zeile den Input und welche den Output darstellen soll. Die Differenz zwischen Output und Input soll aber i. a. nach oben oder nach unten begrenzt werden können. Wird eine Beschränkung nach unten gewünscht, so ist die Inputsicht notwendig, derzufolge die Urbildkomponente als Output und die entsprechende Bildkomponente als Input interpretiert werden. Bei der Inputsicht ist die Differenz zwischen Output- und Inputkomponente schon gleich der entsprechenden

Ergebniskomponente, die durch die Vorgabe stets nach unten beschränkt ist. Soll die Differenz zwischen Output und Input dagegen nach oben begrenzt werden, so ist die Outputsicht einzunehmen, bei der die Urbildkomponente als Input und die entsprechende Bildkomponente als Output gedeutet werden. Die Obergrenze, die von der Differenz zwischen Output und Input beachtet werden muß, ist dann bei Umkehrung ihres Vorzeichens als Unterschranke für das entsprechende Komponentenergebnis in die Planungsungleichung einzusetzen. Die zu wählende Sichtweise hängt also nur davon ab, ob die Differenz zwischen Output und Input in der betrachteten Komponente nach unten oder nach oben beschränkt werden soll. Natürlich kann es erwünscht sein, die Differenz zwischen Output und Input in einer Komponente sowohl nach oben als auch nach unten zu begrenzen. In diesem Fall muß das entsprechende Modellobjekt in zwei Objekte aufgespalten werden, die sich lediglich durch die Sichtweise, unter der sie in das Modell eingeführt werden, unterscheiden und in allen ihren übrigen Merkmale übereinstimmen. Dies muß dann allerdings schon bei der Aufstellung der Aktivitätenfunktion berücksichtigt werden, indem für jede Aktivität, die die Funktion erfüllt, garantiert wird, daß die zu den beiden Modellobjekten gegebenen Input- und Outputgrößen jeweils zueinander konsistent sind. Durch solche Maßnahmen ist es also grundsätzlich möglich, für jedes Modellobjekt die Output/Input-Differenz gleichzeitig nach oben und nach unten zu beschränken, ohne die Gestalt der Planungsungleichung $y \leq t(x)$ zu ändern.

In einigen Fällen ist es notwendig, Output oder Input eines Modellobjektes und nicht nur deren Differenz durch Ober- oder Untergrenzen zu beschränken. Auch hierfür muß ein weiteres Modellobjekt und somit eine neue Zeile in das zwispaltige Aktivitätenschema eingeführt werden. Soll der Input beschränkt werden, so ist dieser in Abhängigkeit von der Sichtweise als Argument- oder Bildkomponente in die neue Aktivitätenzeile einzutragen und die jeweils andere Komponente gleich Null zu setzen. Wird eine Beschränkung des Outputs gewünscht, so

wird analog dazu verfahren. Die Input- und Outputsicht erlauben auch hierbei die Beschränkung des Outputs oder Inputs nach unten bzw. nach oben, wobei die Untergrenzen unverändert und die Obergrenzen nach Umkehrung ihres Vorzeichens in die entsprechende Komponente der Planungsungleichung einzutragen sind. Selbstverständlich können die Ober- und Untergrenzen für Input und Output einer Modellkomponente auf die beschriebene Weise auch gleichzeitig in das Aktivitätenschema eingeführt werden, wie dies schon für ein ursprünglich gegebenes Modellobjekt genutzt werden, so sind hierfür im Aktivitätenschema maximal sechs Zeilen erforderlich, d.h. Modellobjekte, die sich alle auf dasselbe materielle Objekt beziehen und deren Output- und Inputgrößen daher zueinander konsistent gehalten werden müssen.

Literatur: Franke, R.: Betriebsmodelle, Düsseldorf, 1972. Kloock, J.: Betriebswirtschaftliche Input-Output-Modelle, Wiesbaden 1969. Küpper, H.-U.: Dynamische Produktionsfunktion der Unternehmung auf der Basis des Input-Output-Ansatzes, in: ZfB, Jg. 49, 1979. Leontief, W.: Input-Output-Economics, New York, 1966. Nikaido, H.: Convex Structures and Economic Theory, New York, 1968. Pichler, O.: Anwendung der Matrizenrechnung bei der Betriebskostenüberwachung, in: Adam, A. et al. (Hrsg.), Anwendung der Matrizenrechnung auf wirtschaftliche und statistische Probleme, 3. Aufl., Würzburg, 1966. Schumann, J.: Input-Output-Analyse, Berlin, 1968. Steffens, F.: Einplanung von Auftragsnetzen in PPS-Systemen, in: Domschke, W. et al. (Hrsg.), Methods of Operations Research 57, Königstein, 1987a. Steffens, F.: Technische Optimierung und Nettobedarfsbildung in isotonen Input-Output-Systemen, in: Opitz, O., Rauhut, B. (Hrsg.), Ökonomie und Mathematik, Berlin, 1987b. Wenke, K.: Matrizenmodelle in der Großindustrie, in: Adam, A. et al. (Hrsg.), Anwendung der Matrizenrechnung auf wirtschaftliche und statistische Probleme, Würzburg, 1966.

Prof. Dr. F. Steffens, Mannheim

Betriebssystem

B. bilden die Grundlage, auf der Anwendungs- und Dienstprogramme aufbauen. Gelegentlich werden auch die Dienstprogramme, wie Texteditoren, †Compiler, Kommandointerpreter, Sortierprogramme u.ä. dem *B.* zugeordnet. In zunehmendem Maße werden sie jedoch als außerhalb des *B.* liegend betrachtet und der Begriff *B.* auf die gemeinsame Basis von Anwendungs- und Dienstprogrammen eingeschränkt. In dieser Auffassung, die i.w. zugrunde gelegt wird, umfaßt es Programme bzw. Programmteile, die unabhängig von speziellen Anwendungen sind, aber bei vielen Anwendungen benötigt werden.

Ziele, die die Konzeption von *B.* leiten, lassen sich unabhängig von der Verwendung für Monoprozessoren, Multiprozessoren oder †verteilte Systeme in die beiden Klassen Betriebsmittelverwaltung und Programmierfreundlichkeit unterteilen (Sieger, 1988):

- Die *Betriebsmittelverwaltung* ist zuständig für das kontrollierte, kooperative Zusammenwirken der verschiedenen Komponenten einer Rechenanlage. Ein Beispiel für Verwaltungsaufgaben stellt die Koordinierung und Nutzung der Parallelarbeit von †Prozessor und Datenübertragungen zwischen Arbeitsspeicher und peripheren Geräten dar oder die Aufteilung von Plattenspeicherplatz auf verschiedene Anwendungen.

- Unter *Programmierfreundlichkeit* sind Maßnahmen zu sehen, die von Hardwareeinzelheiten abstrahieren und für die Programmierung leichter zu handhabende Konzepte anbieten. Als Beispiel sei das Konzept †Datei mit seinen Lese- und Schreiboperationen erwähnt. Es abstrahiert von Einzelheiten peripherer Geräte und ermöglicht verschiedenste Speicher, sowie Ein- und Ausgabegräte programmtechnisch einheitlich zu handhaben.

Die Norm †DIN 44300 faßt diese Aspekte zusammen und definiert den Begriff *B.* als: "Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften der Rechenanlage die Grundlage der möglichen †Betriebsarten des digitalen Rechensystems bilden und insb. die Abwicklung von Programmen steuern und überwachen."

Für gewöhnlich werden drei *Betriebsmodi* (*Betriebsarten*) unterschieden, die auf breite Anwendungsklassen ausgerichtet sind, nämlich Stapel-, Dialog- und †Echtzeitbetrieb:

- Bei *Stapelbetrieb* (*Batch Processing*) werden die Programme mit allen ihren Eingabedaten und einer vollständigen Ablaufbeschreibung dem Rechensystem übergeben. Vorrangiges Betriebsziel ist die möglichst gute Gesamtnutzung des Rechensystems.

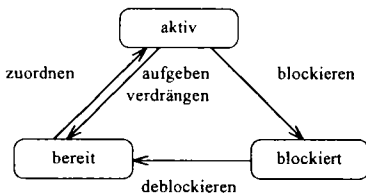
- Bei *Dialogbetrieb* (*Interactive Processing*) kann der Anwender auch während der Programmausführung in das Ablaufgeschehen eingreifen und (abhängig vom Verlauf der bisherigen Ausführung) weitere Eingabedaten bereitstellen. Diese Betriebsart ist insb. erforderlich, wenn die Lösung einer Aufgabe nur schwer im Vorhinein planbar ist, wie bei den meisten Aufgaben der Textverarbeitung oder interaktiven Entwurfsverfahren. Im Gegensatz zum †Stapelbetrieb steht das möglichst behinderungsfreie Arbeiten der Anwender im Vordergrund.

- Bei *Echtzeitbetrieb* (*Real Time Processing*) müssen Reaktionen von Programmen, d.h. Zeitpunkte der Ein-/Ausgaben, in engen Grenzen an die reale Zeit geknüpft werden können. Häufig ist dazu eine starke Einflußnahme auf die Reihenfolge bei der Abwicklung von Programmen erforderlich, damit benötigte Rechenergebnisse rechtzeitig zur Verfügung stehen. Haupteinsatzgebiet dieser Betriebsart ist die †Steuerung und †Regelung technischer Prozesse

Realisieren lassen sich diese Betriebsarten, obwohl sie widersprüchliche Ziele verfolgen, auf der Basis von wenigen Grundkonzepten und daran geknüpften Bearbeitungsstrategien. Traditionelle Beschreibungen gehen vom ablaufenden Programm aus, das sie als Prozeß bezeichnen. Prozesse werden zudem als die †Instanzen betrachtet, denen vom *B.* Betriebsmittel zur Verfügung zu stellen sind. Diese Sichtweise ist geleitet von der Vorstellung eines Monoprozessors, auf dem genau ein Programm abläuft. Mit dem Aufkommen von Multiprozessoren und †verteilten Systemen erweist es sich als unzweckmäßig, Prozesse als nicht weiter gegliederte Verwaltungseinheiten

anzusehen. Eine solche Sicht behindert unnötig stark die einheitliche Behandlung der \uparrow parallelen, \uparrow verteilten Verarbeitung innerhalb einer Anwendung und des Zusammenwirkens von Prozessen in einem gemeinsamen Speicher. Die neuere Entwicklung trennt deshalb die Fähigkeit zur Befehlsabwicklung von der Fähigkeit zur \uparrow Datenspeicherung und betrachtet die Begriffe *Faden* (Thread, Thread of Control), *Adreßraum* (Address Space) und *Datei* (File), sowie die Mechanismen zur *Zugriffskontrolle* (Access Control) und zur *Interaktion* von Fäden (Inter Process Communication) als grundlegend.

Fäden wird die Fähigkeit zugeschrieben, Befehle (*Instructions*) eines Programms in einer vorgebbaren Reihenfolge sequentiell auszuführen. Sie werden vom *B.* durch zeitliches Multiplexen der Verarbeitungseinheiten der Rechenanlage implementiert. Ein Faden befindet sich zu jedem Zeitpunkt in einem der drei Zustände bereit (*Ready*), aktiv (*Running*) oder blockiert (*Waiting*).



Einem aktiven Faden ist eine *Verarbeitungseinheit* zugeordnet, die seine Befehle ausführt. Fäden, für die zum Betrachtungszeitpunkt keine Befehle abgewickelt werden, weil alle Verarbeitungseinheiten zugeordnet sind, gelten als bereit. Fäden, die aus logischen Gründen keine Befehle ausführen sollen, z.B. weil für eine sinnvolle Weiterarbeit erst das Ende eines Eingabevorgangs abgewartet muß, werden als blockiert geführt. Übergänge von aktiv nach blockiert werden dadurch hervorgerufen, daß ein aktiver Faden dem *B.* mitteilt, daß seine Weiterführung bis zum Zutreffen einer vorgebbaren Bedingung aufgeschoben werden soll. Mit der Erfüllung dieser Bedingung wird er in den Zustand bereit versetzt. Übergänge von aktiv nach bereit können erfolgen, indem ein Faden explizit (durch einen *B.*-Aufruf) die Zuordnung der Verarbei-

tungseinheit aufgibt, oder indem das *B.* sie zwangsweise aufhebt und den Faden von der Verarbeitungseinheit verdrängt. Den Belangen der verschiedenen Betriebsarten wird durch die Wahl der Verdrängungszeitpunkte und die Strategie zur Auswahl zuzuordnender Fäden Rechnung getragen. Für *Stapelbetrieb* kommen vorwiegend nicht-verdrängende Zuordnungsstrategien zum Einsatz. Beim *Dialogbetrieb* ist es meist erwünscht, daß keine Fäden bei der Zuordnung stark benachteiligt werden. Fäden werden deshalb nach einer sog. Zeitscheibe von wenigen msec bis sec verdrängt und reihum zugeordnet (Round Robin). Ein für interaktive Nutzer angenehmeres Verhalten wird erzielt, wenn statt dessen zur Zuordnung jeweils ein Faden ausgewählt wird, der die geringste Rechenzeit verbraucht hat. In Systemen für *Echtzeitbetrieb* werden i.a. die Fäden vom Anwender mit einer Priorität (Dringlichkeit) versehen und das *B.* sorgt dafür, daß bereite Prozesse keine höhere Priorität haben als zugeordnete. Manche Systeme lassen dynamische Prioritäten zu, die sich an Eigenschaften der Fäden orientieren, z.B. am Zeitpunkt, bis zu dem die Werte für eine Ausgabe spätestens ermittelt sein müssen.

In einem groben Zeitraster betrachtet entsteht der Eindruck, daß die Fäden gleichzeitig Befehle ausführen. Da dies jedoch bei Betrachtung in einem feineren Zeitraster nicht notwendigerweise so ist, sondern im Gegenteil der exakte Zusammenhang zwischen der Befehlsausführung verschiedener Fäden an der Benutzerschnittstelle nicht angegeben werden kann, wird die oben dargestellte Art der Abarbeitung gleichzeitig existierender Fäden als nebenläufig bezeichnet.

Adreßräume legen den *Adressierungsrahmen* fest, der die Zulässigkeit und Interpretation von Adressen in auszuführenden Befehle bestimmt (Wettstein, 1993). Das Konzept Adreßraum, wie es von Betriebssystemen eingeführt wird, basiert auf den hardwarebezogenen Begriffen Programm- und Speicheradreßraum. Der *Programmadreßraum* besteht aus den Adressen, die der \uparrow Prozessor bei Ausführung von Befehlen als Befehls- oder Datenadressen bilden kann und die für zulässig befunden werden. Der *Spei-*