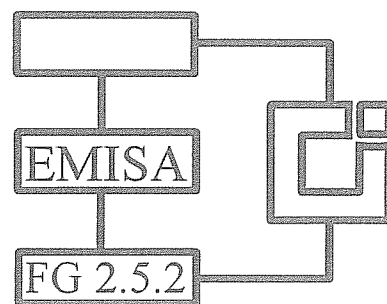


EMISA FORUM



Mitteilungen der GI-Fachgruppe "Entwicklungsmethoden für Informationssysteme und deren Anwendung"

Heft 2

1995

Inhalt:

| | |
|--|----|
| Editorial..... | 1 |
| Aus der Fachgruppe..... | 3 |
| Fachbeiträge | |
| M. Bertram: Qualitätsaspekte der Wiederverwendung von Schemata | 18 |
| S. Conrad, G. Denker, M. Gogolla, R. Herzig, N. Vlachantonis, H.-D. Ehrich: Entwicklung zuverlässiger Informationssysteme | 25 |
| G. Englmeier: Boolesch Algebraische Generalisierung/Spezialisierung von Objekttypen | 34 |
| E. Ortner: Natürlichsprachlicher Entwurf von Informationssystemen | 47 |
| H. Paul: Modellierung in soziotechnischen Systemen - Von Menschen, Organisationen, Modellierern und Modellen | 66 |
| Arbeitskreis "Grundlagen der Modellierung und Ausführung von Workflows"..... | 77 |
| Tagungsberichte | 79 |
| Buchbesprechungen | 80 |
| Ankündigungen | 82 |

Impressum

Das Mitteilungsblatt der Fachgruppe 2.5.2 "Entwicklungsmethoden für Informationssysteme und deren Anwendung (EMISA)" im Fachausschuß 2.5 der Gesellschaft für Informatik (GI) erscheint zweimal jährlich, und zwar im Januar und im August. Es wird den Mitgliedern der Fachgruppe zugesandt. Mitglied kann jeder werden, der sich für die Zielsetzungen der Fachgruppe interessiert und sich mit einer stichwortartigen Angabe seiner Arbeitsgebiete anmeldet. Es wird ein jährlicher Mitgliedsbeitrag von 15 DM (Nicht-GI-Mitglieder 30 DM) erhoben.

Durch das Mitteilungsblatt sollen aktuelle Informationen unter den Mitgliedern der Fachgruppe ausgetauscht werden. Die Herausgeber bitten daher alle Leser, sich möglichst rege an der Gestaltung des Blattes zu beteiligen.

Es können für die Fachgruppe relevante Beiträge unter anderem zu folgenden Rubriken eingereicht werden:

- Fachbeiträge
- Berichte aus der Praxis
- Projektberichte (auch über laufende Projekte)
- Vorstellung von Arbeitsgruppen
- Buchbesprechungen
- Aktuelle Publikationen, Dissertationen, Habilitationen
- Tagungsberichte
- Call for Papers, Einladungen, Programme
- Leserbriefe, E-Mail-Zuschriften

Mit der Zusendung eines Beitrags an einen der Herausgeber (die Adressen werden auf der dritten Umschlagseite angegeben) ist das Einverständnis zur Veröffentlichung im Mitteilungsblatt verbunden. Jeder Beitrag wird ohne Begutachtung veröffentlicht (wenn er dem Themenspektrum der EMISA entspricht), er gibt nur die individuelle Meinung des Autors wieder. Die Beiträge können in deutscher oder englischer Sprache abgefaßt sein. Sie sollten keine Seitennumerierung enthalten. Dringende Mitteilungen können mittels E-Mail an die Redaktion (oberweis@aifb.uni-karlsruhe.de) geschickt werden.

Redaktionsschluß für die Januar-Ausgabe ist der 5. Dezember, für die August-Ausgabe der 7. Juli (wenn nicht anders angekündigt).

Herausgeber: Leitungsgremium der Fachgruppe EMISA

Auflage: 2300

Ergänzend zum EMISA FORUM gibt es die Möglichkeit, unter World Wide Web Informationen mit Relevanz für die Fachgruppe bereitzustellen (<http://sunsite.informatik.rwth-aachen.de/Societies/GI-EMISA/>; Ansprechpartner: Manfred Jeusfeld, jeusfeld@usthk.ust.hk).

EDITORIAL

Liebe EMISA-Mitglieder,

es gibt wieder über einige Dinge zu berichten, was ich an dieser Stelle in aller Kürze tun möchte. Die Fachgruppenleitung hat sich auf ihrer letzten Sitzung einige strategische Gedanken gemacht, insbesondere darüber, was wir (momentan) für **die aktuellsten Themen** für die EMISA halten und wir daher z.B. im Rahmen von Workshops, Beiträgen im Forum oder Fachgruppentreffen behandeln sollten. Jedes FGL-Mitglied hat die von uns zusammengetragenen Themen mit einer Priorität versehen, und ich habe hieraus unsere derzeitige "Hitliste" ermittelt. Diese lautet wie folgt:

1. Modellierung in soziotechnischen Systemen
2. (a) Workflow-Management, (b) Objekt-Orientierung (in Analyse und Entwurf)
3. Qualitätsgewährleistung und -sicherung (z.B. ISO 900x)
4. Ausbildung für den Informationssystem-Entwurf
5. (a) Metamodelle und Metamodellierung, (b) Entwurf und Entwicklung verteilter, insbes. föderativer Informationssysteme
6. Natürlichsprachlicher Entwurf
7. Geschäftsprozeßmodellierung und Reengineering
8. Entwicklung von Multimedia-Anwendungen
9. Formale Spezifikation von Informationssystemen
10. (a) Informationssysteme im Internet, (b) Requirements Engineering
11. Management von Informationssystem-Entwicklungsprojekten
12. Methodeneinsatz in der Praxis
13. Informationssystem-Architekturen
14. Benutzer-, Kundenpartizipation in der Softwareentwicklung
15. Vorgehensmodelle
16. Entwicklung von Applikationssoftware mit (Normteile-) Katalogen
17. Systementwicklung im Umfeld von Standardsoftware
18. Customizing von Standard-Software
19. Implementierungstechniken
20. Angemessenheit von Informationssystem-Einsatz
21. Soziale Voraussetzungen und Folgen des Einsatzes von Informationssystemen

- 22. Teamunterstützung bei der Informationssystem-Entwicklung
- 23. CASE-Werkzeuge
- 24. Theoretische Grundlagen von Workflow-Modellen

Diese Liste ist nun keineswegs so zu verstehen, daß wir die genannten Themen chronologisch abarbeiten wollen; sie bedarf sicher einiger Verfeinerungen sowie einer gewissen Strukturierung. Zunächst wollen wir sie zur Diskussion stellen. **Wir würden uns daher über Feedback von Ihnen freuen**, etwa Ergänzungen, Ihre persönliche Priorisierung dieser Themen oder einfach ein Kommentar, denn sonst gehen wir für die nahe Zukunft davon aus, daß wir hiermit auch Ihr Interesse getroffen haben bzw. bei entsprechenden Aktivitäten treffen werden. Sehr begrüßen würden wir u.a. Praxisberichte zu den genannten Themen, denn die Universitätsangehörigen unter uns interessiert z.B. die Frage, ob wir denn auch angemessen ausbilden.

Es ist allerdings so, daß wir zu einigen der o.g. Themen (insbesondere sogar zu den höchstplatzierten) bereits heute etwas zu sagen haben. Über *Modellierung in soziotechnischen Systemen* lesen Sie in diesem Forum einen Bericht von Dr. Paul zum ersten Treffen der gleichnamigen Arbeitsgruppe. Zum Thema *Workflow-Management* wurde vor kurzem ein Arbeitskreis im Fachausschuß 2.5 gegründet, der sich Ende Juni in Ulm mit unserem FGL-Mitglied Stefan Jablonski als Sprecher konstituiert hat. Dank dieser Personalunion werden Sie über diesen AK des öfteren im Forum lesen können. Thema 11, *Informationssysteme im Internet*, wollen wir zum Thema des 96er Fachgruppentreffens machen, das Manfred Jeusfeld in Aachen veranstalten wird.

In diesem Zusammenhang noch ein Hinweis auf unsere neueste technische Errungenschaft: **EMISA goes on-line!** Es hat sich vielleicht inzwischen herumgesprochen, daß die GI beabsichtigt, einen WWW-Server einzurichten. Derzeit existiert ein nicht-offizieller derartiger Server bereits in Freiburg:

<http://www.iig.uni-freiburg.de/gi/gi-online.html>

Dort finden Sie inzwischen auch einen Eintrag der EMISA. Genauere Informationen hierzu gibt Ihnen Manfred Jeusfeld an anderer Stelle in diesem Heft. Der "übliche" Zugang zu einem WWW-Server erfolgt über einen Internet-Anschluß sowie einen WWW-Browser (z.B. NCSA Mosaic, das per anonymem FTP von <ftp.ncsa.uiuc.edu> erhältlich ist, oder Netscape).

Ich möchte nicht versäumen, auch an dieser Stelle noch einmal explizit auf unser diesjähriges **Fachgruppentreffen** im Oktober in Karlsruhe hinzuweisen (diesmal ohne MobIS, aber zeitlich so liegend, daß Sie auch am MobIS-Treffen teilnehmen können); das Programm findet sich an anderer Stelle in diesem Heft. Ich würde mir wünschen, daß wir den Teilnehmererfolg vom letzten Jahr wiederholen können. Im Rahmen des Fachgruppentreffens wird es wie im letzten Jahr eine **Mitgliederversammlung** geben, zu der ich Sie an dieser Stelle herzlich einladen möchte.

Das vorliegende Heft enthält die üblichen Rubriken. Ich wünsche Ihnen viel Spaß beim Lesen, einen schönen Sommer und hoffe, viele von Ihnen in Karlsruhe begrüßen zu können.

Münster, im Juli 1995

Gottfried Vossen

EMISA online - aktuelle Informationen für und über die Fachgruppe

Manfred A. Jeusfeld

Hong Kong University of Science and Technology, ISMT, Sai Kung, HK
jeusfeld@usthk.ust.hk

Schlägt man in diesen Tagen eine Fachzeitschrift oder den Wirtschaftsteil einer Zeitung auf, so stößt man mit einiger Regelmäßigkeit auf Artikel über die Datenautobahn, auch *information highway* oder *Infobahn* genannt. In der Tat hat im Laufe der letzten Jahren die Nutzung des weltweiten Rechnernetzes durch das WWW-Paradigma (*worldwide web*) explosionsartig zugenommen. Die Vorteile liegen auf der Hand: das Rechnernetz ist vorhanden, Bereitstellung von sowie Zugriff auf Dokumente erfordern wenig Aufwand, und die notwendigen Programme sind für praktisch alle Rechnerarchitekturen (PC, Mac, Unix, Mainframe) verfügbar.

Diesen Argumenten wollte sich die Fachgruppenleitung nicht verschließen: seit Mai 1995 ist EMISA mit einem zwar bescheidenen aber stetig wachsendem Angebot im WWW vertreten. Falls Sie Zugang zum WWW haben, so können sie *EMISA online* unter der Adresse¹

<http://sunsite.informatik.rwth-aachen.de/Societies/GI-EMISA/> zugreifen. Die Programme Mosaic, Lynx oder Netscape sind problemlos verwendbar. Folgende Rubriken bilden das Gerüst von *EMISA online*:

- **EMISA-FORUM:** Hier finden Sie u.a. Inhaltsverzeichnisse der vergangenen Hefte des EMISA-FORUMs und Vorschauen auf die nächste Ausgabe. Die Rubrik wird von Andreas Oberweis, Redaktion EMISA-FORUM, zusammengestellt.
- **Fachgruppentreffen:** Diese Rubrik informiert Sie über unsere jährlichen Fachgruppentreffen. Wenn technische Fragen geklärt sind, werden Sie über diese Seite auch ausgewählte Artikel der Fachgruppentreffen zugreifen können.
- **Arbeitstreffen:** Diese Liste gibt ein Vorschau auf die geplanten Arbeitstreffen mit EMISA-Beteiligung. Zudem können Sie sich hier über interessante Veranstaltungen außerhalb der EMISA auf den neusten Stand bringen.
- **Externe Veranstaltungen:** Diese Liste soll Ihnen bei der Suche nach Informationen über Konferenzen, Seminare etc. mit Bezug zu dem Thema unserer Fachgruppe helfen. Die Liste ist in die Bereiche Datenbanken, Informationssysteme, Software Engineering und Sonstiges unterteilt. Weitere Bereiche mögen folgen.
- **Mitteilungen für Mitglieder:** Haben Sie eine wichtige Nachricht für Mitglieder der EMISA? Möchten Sie etwa auf die Gründung eines Arbeitskreises hinweisen? Diese Rubrik ist das elektronische 'schwarze Brett' der Fachgruppe.

Die Dokumente werden, so dies mit vertretbarem Aufwand möglich ist, auf dem aktuellen Stand gehalten. *EMISA online* ist auf Feedback programmiert: die Rubriken enthalten Elemente, mit denen Sie Ihre Beiträge, etwa die Ankündigung eines Seminars, in das Medium einspeisen können. Dieser interaktive Charakter soll in Zukunft noch gestärkt werden, um gleichzeitig die Attraktivität des Angebots zu steigern und den redaktionellen Aufwand zu begrenzen. *

¹ Wir danken dem Lehrstuhl Informatik V in Aachen für die Bereitstellung von Ressourcen auf dem WWW-Server Sun SITE Central Europe.

E-Mail-Liste

Seit Juli 1995 hat die GI-Fachgruppe 2.5.2 EMISA "Entwicklungsmethoden für Informationssysteme und deren Anwendung" eine E-Mail-Liste eingerichtet. Hierüber können Mitteilungen von allgemeinem Interesse für EMISA-Mitglieder an diejenigen EMISA-Mitglieder, die sich in die Liste eingetragen haben, verteilt werden.

Insbesondere ist an Ankündigungen und Programme von Tagungen, Workshops und Arbeitsgruppen gedacht, aber auch an andere Beiträge, die ins EMISA-Forum passen würden. Um das E-Mail-Volumen zu beschränken, sollten allerdings anstelle langer Mitteilungen nur Verweise auf die Originaltexte, z.B. WWW- oder FTP-Adressen, angekündigt werden.

Benutzung:

- Die Adresse des E-Mail-Verteilers lautet: `emisa@informatik.uni-hannover.de`
- Zur Aufnahme in den Verteiler ist an `majordomo@informatik.uni-hannover.de` eine Mail zu senden, die im Rumpf eine Zeile `subscribe emisa` enthält (nicht als Subject-Zeile); die E-Mail-Adresse wird dann aus dem Header der Mail übernommen. Alternativ kann die E-Mail-Adresse mit `subscribe emisa <Adresse>` explizit angegeben werden.
- Eine Streichung aus dem Verteiler erfolgt analog mit: `unsubscribe ...`
- Informationen über weitere Kommandos können mit der Rumpfzeile `help` von `majordomo@informatik.uni-hannover.de` angefordert werden.
- Andere Anfragen bitte ich an `emisa-owner@informatik.uni-hannover.de` zu richten.
- Ein Archiv mit den letzten Mails findet sich im WWW unter der URL
`http://www.informatik.uni-hannover.de/emisa`
ist aber auch von der WWW-Startseite der Fachgruppe erreichbar.

Das Kleingedruckte: Wiederholungen von Mails sollten möglichst vermieden werden; für zweite Tagungsankündigungen reicht oft eine kurze Erinnerung. Nicht erwünscht sind Produktwerbung und Stellenausschreibungen. Die Liste ist auch leider nicht als Diskussionsforum gedacht, denn dafür sind direkte E-Mail und News geeignetere Instrumente. — Bis auf weiteres wird diese Liste moderiert, um ggf. diese Richtlinien aufgrund von Erfahrungen zu überarbeiten. Dadurch können sich Verzögerungen beim Versand ergeben.

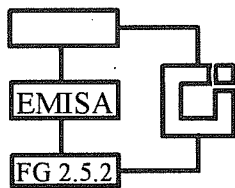
Udo Lipeck

GI-Fachgruppe 2.5.2 EMISA "Entwicklungsmethoden für Informationssysteme und deren Anwendung"

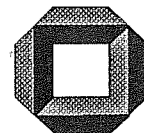
Einladung und Programm

EMISA - FACHGRUPPENTREFFEN 1995 "REQUIREMENTS ENGINEERING FÜR INFORMATIONSSYSTEME"

12.-13. Oktober 1995



Universität
Karlsruhe



Institut für Angewandte Informatik und Formale
Beschreibungsverfahren

Veranstaltungsort: Gaede-Hörsaal

Requirements Engineering ist die ingenieurmäßige Sammlung, Beschreibung und Analyse von Anforderungen an DV-Systeme. Im Rahmen des EMISA-Fachgruppentreffens 1995 sollen unter anderem folgende Aspekte diskutiert werden:

- Erhebung und Analyse von Anforderungen,
- Beschreibungssprachen für Anforderungen,
- formale vs. informale Anforderungsbeschreibungen,
- methodenbezogene Fragestellungen,
- Verwaltung und Dokumentation von Anforderungen,
- Werkzeuge,
- Qualität und Effizienz,
- Objektorientierung.

Donnerstag, 12. Oktober:

10:00 - 10:15 **Begrüßung**

10:15 - 12:15 Sitzung 1: Objektorientierung

Rainer Burkhardt, Martin Wolf, Ilka Philippow (TU Ilmenau):
Objektorientierte Modellierung mit dem Objekt-Prozeß-Modell

Daniel Moldt (Uni Hamburg):
Extensions of normal OOA methods and techniques for distributed Real-Time Systems

Barbara Röllnreiter, Ralph Penno (Softlab, München):
Ansatz zur Kombination von objektorientierten Methoden für die Fachkonzeptphase

Michael Rohloff (München):
Analyse und Prozeßentwurf betrieblicher Abläufe im OGM-Ansatz

12:15 - 13:45 **Mittagessen**

13:45 - 15:15 Sitzung 2: Spezielle Anwendungsbereiche

Brigitte Baldi, Werner Brettreich-Teichmann, Karin Gräslund, Rainer Hofmann, Peter Konrad, Helmut Krcmar, Joachim Niemeier, Gerhard Schwabe, Dietrich Seibt (Uni Hohenheim, Uni Köln, KPMG Frankfurt, Fraunhofer-IAO Stuttgart, BIFOA Köln):
BTÖV - eine Methode für die bedarfsgerechte Gestaltung von Telekooperation in der öffentlichen Verwaltung

Rainer Perkuhn (Uni Karlsruhe):
Requirements Engineering in MIKE

Mathias Philipp (Uni Frankfurt):
Aus Handels- und Steuerrecht abgeleitete Anforderungen an den Entwurf von Geschäftsprozeßmodellen und Workflow-Anwendungen

15:15 - 15:45 **Kaffeepause**

15:45 - 17:15 Sitzung 3: Formale Modellierung

Udo W. Lipeck, Stefan Brass (Uni Hannover):
Defaults in formalen Spezifikationen von Systemanforderungen

P. Wendorff (Duisburg):
Implementierung einer anwendungsspezifischen formalen Beschreibungssprache durch einen Application-Framework

Georg V. Zemanek, Hans W. Nissen (RWTH Aachen):
Formales System und Vorgehensweise zur Modellierung von Geschäftsprozessen

17: 30- 18:30 **EMISA-Mitgliederversammlung**

20:00 **gemeinsames Abendessen**

Freitag, 13. Oktober:

09:00 - 10:30 Sitzung 4: Werkzeuge

O.J. Bott, O.-S. Penger, A. Terstappen (Uni Hildesheim):

Ein Ansatz zur Methoden- und werkzeuggestützten Anforderungs- und Systemspezifikation auf der Grundlage objektorientierter Modellierungs- und Simulationstechniken

D. Moldt (Uni Hamburg):

Anforderungen an Petrinetzwerkzeuge für den kommerziellen Einsatz

Klaus Pohl, Ralf Dömges, Peter Haumer, Ralf Klamma, Klaus Weidenhaupt (RWTH Aachen):

PRO-ART: Process and Repository based Approach to Requirements Traceability

10:30 - 11:00 Kaffeepause

11:00 - 12:30 Sitzung 5: Methodenbezogene Fragestellungen

Peter Jaeschke (PROMATIS Informatik, Karlsbad):

Entity-Relationship-Modell-Clustering

Christian Kop, Heinrich C. Mayr (Uni Klagenfurt):

Objektorientierte Analyse und konzeptueller Vorentwurf

H. Thoma (Ciba-Geigy, Basel):

Schwachstellen des Requirements Engineering für Informationssysteme in der Praxis

12:30 - 13:45 Mittagessen

13:45 - 15:15 Sitzung 6: Qualitäts- und Effizienzaspekte

Angelika Franzke, Andreas Winter (Uni Koblenz-Landau):

Software-Evaluation mit Mitteln des Requirements-Engineering

Georg Herzwurm (Uni Köln):

Kundenorientierte Planung von Softwareprodukten und -prozessen mit Quality Function Deployment

Matthias Riebisch (TU Ilmenau):

Fortschritte im Requirements Engineering als Beitrag zu Kundenorientierung und zur Effizienzverbesserung im Entwicklungsprozeß von Informationssystemen

15:15 - 15:30 Resümee

Programmzusammenstellung:

M. Jeusfeld (Hong Kong University of Science and Technology)

R. Kaschek (Universität Klagenfurt)

C. Kop (Universität Klagenfurt)

A. Oberweis (Universität Karlsruhe)

R. Thurner (Universität Klagenfurt)

G. Vossen (Universität Münster)

Dr. Andreas Oberweis
Universität Karlsruhe
Institut für Angewandte Informatik und
Formale Beschreibungsverfahren
Kollegiengebäude am Ehrenhof

D 76128 Karlsruhe

Tel. 0721-608-4283, Fax 0721-693717
Email oberweis@aifb.uni-karlsruhe.de

**Anmeldung zum EMISA - FACHGRUPPENTREFFEN 1995
"REQUIREMENTS ENGINEERING FÜR INFORMATIONSSYSTEME"**

Universität Karlsruhe, 12.-13. Oktober 1995

Titel, Name, Vorname:

Firma/Institution:

Straße:

PLZ: Ort:

Telefon: Fax: E-Mail:

Verbindl. Anmeldung zum gemeinsamen Abendessen am 12. Oktober: ja nein

Datum: Unterschrift:

**Bitte bis 25. September 1995 an die oben angegebene Adresse
schicken oder faxen!**

(Die Teilnahme am Fachgruppentreffen ist für Fachgruppenmitglieder kostenlos.
Von Nicht-Fachgruppenmitgliedern wird eine Teilnahme-Gebühr in Höhe von 50
DM erhoben. Eine genaue Wegbeschreibung wird nach Eingang der Anmeldung
an die Teilnehmer des Treffens verschickt.)

GI-Fachgruppe 2.5.1 "Datenbanksysteme"

GI-Fachgruppe 2.5.2 "Entwicklungsmethoden für Informationssysteme und deren Anwendung" (EMISA)

SI-Fachgruppe "Data Bases - Theory and Application" (DBTA)

Fachgespräch

"Entwurf und Entwicklung verteilter Informationssysteme"

im Rahmen der gemeinsamen Jahrestagung der Gesellschaft für Informatik (GI) und der Schweizerischen Informatiker Gesellschaft (SI)

Zürich, 18.-20. September 1995

Der anhaltend starke Hardware-Preisverfall mit einem noch nicht dagewesenen Preis-/Leistungsverhältnis von Rechnern, mit einer Leistungsfähigkeit von PC's und Workstations, die in den Mainframe-Bereich hineinreicht, der Trend zu aufwendigen graphischen Benutzeroberflächen sowie der Wunsch nach einem autonomen Arbeiten im Verbund machen es möglich, in verteilten Informationssystemen gemeinsame Aufgaben zu bewältigen. Beispiele hierfür sind das rechnergestützte Zusammenarbeiten von Mitgliedern innerhalb einer Arbeitsgruppe, die abgestimmte Logistik multinationaler Unternehmen in Bereichen wie Lagerbewirtschaftung, Produktion oder Vertrieb sowie die weltweite, koordinierte Produkt-Forschung und -Entwicklung in unterschiedlichen F&E-Zentren.

Trotz der verlockenden technischen Möglichkeiten sind verteilte Informationssysteme in der Praxis bisher nur selten realisiert. Grund hierfür ist z.B. die Komplexität des Entwurfs, des Aufbaus und des Betriebs solcher Systeme. Außerdem verlangt die große Vielfalt an Realisierungsvarianten und an kommerziell verfügbaren Produkten, welche im Rahmen einer verteilten Umgebung zum Einsatz kommen können, die Bereitstellung entsprechenden Wissens für den Einsatz eines verteilten Systems.

Das Ziel des Fachgesprächs ist es, Aspekte des Entwurfs, der Entwicklung und des Betriebs von verteilten Informationssystemen kritisch zu durchleuchten. Neben Konzepten, Lösungen, Lösungsansätzen und Erfahrungsberichten sollen offene Probleme diskutiert werden.

Sitzung FG 8.1:

Montag, 18. September 1995, 13.45 - 15.00 Uhr

1. Georg Reichwein (Fa. Quantum, Dortmund):
Verteilte betriebswirtschaftliche Standardsoftware in heterogenen Umgebungen
2. Volker Sanger, Wolfgang Weitz (Uni Karlsruhe):
Entwurfsvalidation fur verteilte Informationssysteme mit dem graphischen, mehrbenutzerfahigen Pr/T-Netz-Simulator GAPS+
3. Eberhard Stickel (Uni Frankfurt/Oder):
Ein Ansatz zur Beurteilung der Wirtschaftlichkeit verteilter IV-Systeme

Sitzung FG 8.2:

Montag, 18. September 1995, 15.30 - 16.45 Uhr

1. Michael Merz, Kay Muller-Jones, Winfried Lamersdorf (Uni Hamburg):
Mobile Klienten: Ortsubergreifender Zugang zu Diensten in offenen verteilten Informationssystemen
2. Uta Storl, Klaus Kuspert (Uni Jena):
Interoperabilitat zwischen verschiedenen DFR-Anwendungen in einer Client-Server-Umgebung: Konzepte, Realisierung und Bewertung einer ObjectStore-basierten Losung
3. Peter Szczurko (RWTH Aachen):
Integration verteilter heterogener Datenbanken in einer foderativen Qualitatsmanagementarchitektur

Kontaktadresse Tagungsorganisation:

MCI Travel Zurich
Weinbergstrae 11
CH-8001 Zurich

Petrinetze und Informationssysteme in der Praxis

Peter Langner, INNOBIS Prof. Dr. Gerd Scheschonk, C.I.T.

Am 3.04.1995 fand die dritte Tagung der Arbeitsgruppe "Petrinetze und Informationssysteme in der Praxis" im Hause der C.I.T. in Berlin statt. Diese interdisziplinäre Arbeitsgruppe, trifft sich zweimal im Jahr zum Erfahrungsaustausch. Eingeladen hatten die Fachgruppen "Petrinetze" (FG 0.0.1) und "Entwicklungsmethoden für Informationssysteme und deren Anwendungen" (EMISA, FG 2.5.2) der Gesellschaft für Informatik (GI).

In einem Vortrag von Prof. Dr. Scheschonk wurde die Modellierung eines Dokumentenverwaltungssystems in Form eines gefärbten Netzes vorgestellt. Dieses System wurde im Rahmen eines Projektauftrags erstellt. Ziel des Projektes war es, Anfragen nach Dokumenten innerhalb einer vorgeschriebenen Zeit zu erfüllen. Hierzu mußte eine Hardwarekonfiguration ermittelt werden, die den Zugriff auf ca. 20 Millionen räumlich verteilte Dokumente gewährleistet. Zur Modellierung und Simulation wurde das Werkzeug Design/CPN verwendet. (vgl. G. Scheschonk, M. Timpe: Simulation and Analysis of a Document Storage System. Proceedings of the 15th International Petri Net Conference, Springer Verlag 1994) In der Diskussion zum Vortrag wurde insbesondere die Frage erörtert, wie sich aus der Aufgabenstellung das zugehörige Netz ableiten läßt. Es wurde eine Abwägung vorgenommen zwischen der Darstellung einer kompakten Netzstruktur mit entsprechend komplexen Beschriftungen und einer Darstellung, die auf Beschriftungen eher verzichtet und entsprechend umfangreicher in der Netzstruktur ist.

Im Anschluß an den Vortrag stellten die 18 Teilnehmer aus Forschung und Wirtschaft in drei Kleingruppen den Stand ihrer aktuellen Projekte, in denen sie Petrinetze einsetzen dar, bzw. welches ihre Interessen an einem Einsatz von Petrinetzen sind. Die Themen wurden im Plenum zusammengetragen und bilden das Programm für die nächsten Treffen. Es sind dies im einzelnen:

1. Die Behandlung von Zeiten in Netzen
2. Simulation

Hierbei soll insbesondere eingegangen werden auf Simulationskonzepte (Was kann modelliert werden?), Softwarewerkzeuge zur Simulation (Tips & Tricks), Qualität der Simulationsergebnisse und Hardwarekonfigurationen für Simulationsumgebungen (z.B. der Einsatz von Simulationsservern).

3. Modellierung von Geschäftsprozessen (Workflow)
4. Werkzeuge

Schwerpunkthemen sind hierbei Austauschformate für Softwarewerkzeuge und die Offenheit der Systeme.

5. Benutzerschnittstellen

Im Zusammenhang mit Benutzerschnittstellen stellen sich u.a. die Fragen, welches die möglichen Benutzergruppen sind und welchem Benutzertyp man welche Art von Benutzeroberfläche zumuten kann.

6. Petrinetze im Rahmen eines Systementwicklungsprozesses

Welche Methoden verwendet man für den Einsatz von Petrinetzen?

Das nächste Treffen der Arbeitsgruppe wird am 9. Oktober 1995, von 9.30h bis 17.00h in Bonn im Hause der Bonndata GmbH stattfinden. Das Thema des Treffens sind die

Vorgehensweisen beim Einsatz von Petrinetzen

Vorgesehen sind am Vormittag zwei Vorträge:

D. Moldt: "Optionen für die Systemspezifikation mit Petrinetzen", Universität Hamburg und

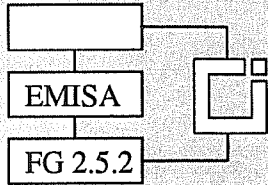
Dr. H. Bierfert: "Erfahrungen beim Einsatz von RFA-Netzen", Bonndata GmbH

Am Nachmittag soll in Kleingruppen eine Klassifizierung von Petrinetz-Projekten, Zielgruppen, Umfeldern sowie bekannter Vorgehensweisen vorgenommen werden.

Zum Treffen und zur Mitarbeit in der Arbeitsgruppe eingeladen sind alle, die Petrinetze im Kontext von Informationssystemen einsetzen oder einsetzen möchten, um z.B. Geschäftsprozesse zu analysieren, interaktive Systeme benutzerorientiert zu beschreiben, Informationssysteme zu modellieren und zu dokumentieren, organisatorische Abläufe zu gestalten, etc.

Zur Anmeldung für das nächste Treffen und für Rückfragen wenden Sie sich bitte bis zum 8. September 1995 an:

Herrn Peter Langner
INNOBIS Unternehmensberatung und Software GmbH
Holsteiner Chaussee 183b
D-22457 Hamburg
Tel. +49 (0) 40 55 9876-0
Fax. +49 (0) 40 55 9876-99



Call for Papers

Natürlichsprachlicher Entwurf von Informationssystemen
- Grundlagen, Methoden, Werkzeuge, Anwendungen -

Workshop, 28.-30. Mai 1996, Ev. Akademie Tutzing

In manchen Software-Unternehmen ist inzwischen der diagrammunterstützte Entwurf von Anwendungssystemen um einen methodisch-natürlichsprachlichen Beginn der Entwicklungsarbeit ergänzt worden. An Universitäten und Forschungseinrichtungen gibt es Projekte, die den Entwurf linguistischer Methoden und die Entwicklung von Werkzeugen zur Unterstützung natürlichsprachlicher Systementwicklung zum Ziele haben.

Die GI-Fachgruppe 2.5.2 EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung) wird einen „Pfingst-Workshop“ zu diesem Themenkomplex in der Evangelischen Akademie Tutzing veranstalten. Mit dem Workshop ist geplant, eine Diskussion innerhalb und zwischen Teildisziplinen wie Requirements Engineering, Information Systems Engineering, Datenbankentwurf, Künstliche Intelligenz, (Computer-)Linguistik und Wissenschaftstheorie anzuregen. Erwünscht sind Beiträge zu folgenden Themenschwerpunkten:

- Grundlagen methodisch-natürlichsprachlicher Informationssystementwicklung
- Entwicklung einzelner Methoden und ihre Integration in Vorgehensmodelle
- Werkzeugentwicklungen und Werkzeugeinsatz
- Projekte bei Anwendern sowie in Software-Unternehmen

Es ist geplant, eine Mischung aus forschungs- und anwendungsorientierten Beiträgen zusammenzustellen, welche in einem Tagungsband publiziert und den Teilnehmern vor Workshop-Beginn vorliegen werden. Beiträge im Umfang von 15 DIN A4 Seiten (1,5-zeilig) sind in vierfacher Ausfertigung einzusenden an:

Prof. Dr. Erich Ortner
Lehrstuhl für Informationsmanagement
Universität Konstanz, Fachgruppe Informationswissenschaft
Postfach 5560 D87/88, 78434 Konstanz

Wichtige Termine:

- bis 15.11.95 Einreichung eines Beitrags
- bis 20.12.95 Benachrichtigung über Annahme/Ablehnung
- bis 15.03.96 Abgabe der Endfassung

Organisationskomitee

E. Ortner, Universität Konstanz
H. Thoma, Ciba-Geigy Basel

Programmkomitee

C. Eckardt, Software AG Darmstadt
W. Hesse, Universität Marburg
S. Jablonski, Universität Erlangen-Nürnberg
U. Lipeck, Universität Hannover
H.C. Mayr, Universität Klagenfurt
E. Ortner, Universität Konstanz (Vorsitz)
K. Pohl, RWTH-Aachen
D. Steinbauer, GEZ Köln
R. Studer, Universität Karlsruhe
M. Seubert, SAP Walldorf
B. Thalheim, Technische Universität Cottbus
H. Thoma, Ciba-Geigy Basel
A M. Tjoa, Technische Universität Wien

Weitere Informationen

Bruno Schienmann
Lehrstuhl für Informationsmanagement
Universität Konstanz, Fachgruppe Informationswissenschaft
Postfach 5560, 78434 Konstanz

Telefon: 07531/88-3098

Email: schien@inf-wiss.uni-konstanz.de

GI-Fachgruppe 2.5.2 EMISA "Entwicklungsmethoden für
Informationssysteme und deren Anwendung"

EMISA-FACHGRUPPENTREFFEN 1996

- Vorläufiger Aufruf -

"Informationsserver für das Internet: Anforderungen, Konzepte, Methoden"

10.-11. Oktober 1996 in Aachen

Das *Internet* ist die weltweit akzeptierte Plattform für Informationsaustausch in Rechnernetzen. Diverse Einrichtungen bieten große Informationssammlungen im Internet an (Informationsserver) -- häufig im populären WWW-Format. Herkömmliche Entwurfsmethoden sind u.a. wegen der Vielfalt der Dokumente bedingt brauchbar. Über die Nutzer eines Informationsservers ist wenig bekannt. Sind also Chaos und Wildwuchs unvermeidlich? Auf dem Fachgruppentreffen soll diese Frage aus der Entwurfsperpektive diskutiert werden. Blickwinkel sind unter anderem:

- a. Modellierungs- und Entwurfsmethoden für Informationsserver im Internet
- b. Entwurf von Benutzerschnittstellen zu Informationsservern
- c. Verpackung konventioneller Informationssysteme als Informationsserver
- d. Modelle (wissenschaftlicher) Kooperation mit Hilfe von Informationsservern
- e. Fallstudien über die Einführung und Nutzung von Informationsservern im Internet
- f. Kopplung autonomer Informationsserver zur Formung höherwertiger Angebote
- g. Modellierung von Beziehungen zwischen verteilten Informationsservern
- h. Konzepte für Datenschutz/-sicherheit bei allgemein zugänglichen Informationsservern

Die Beiträge werden auf inhaltlichen Bezug zum Thema geprüft. Einreichungen aus der Praxis sind besonders willkommen. Die akzeptierten Beiträge werden im EMISA FORUM Heft 1/1997 sowie im Internet veröffentlicht. Aktuelle Informationen zum Fachgruppentreffen finden Sie im WWW unter der Adresse

[http://SunSITE.Informatik.RWTH-Aachen.DE
/Societies/GI-EMISA/FG-Treffen/1996/](http://SunSITE.Informatik.RWTH-Aachen.DE/Societies/GI-EMISA/FG-Treffen/1996/)

Termine:

- 31.5.1996: Einreichung von Beiträgen (500-2000 Worte)
1.7.1996: Benachrichtigung über Annahme bzw. Ablehnung
14.9.1996: Abgabe der endgültigen Beiträge

Bitte senden Sie Ihre Beiträge in dreifacher Ausfertigung an:

Manfred Jeusfeld, c/o Informatik V, RWTH Aachen, D-52056 Aachen
Fax: 0241-8888-321, Email: jeusfeld@informatik.rwth-aachen.de

Email-Einreichungen in Standard-Postscript werden gern entgegengenommen. Eingereichte Beiträge müssen auf der Titelseite Namen und Adressen der Autoren enthalten. Die Autoren werden gebeten, die Einordnung ihres Beitrags die obigen Kategorien a-h auf der Titelseite kenntlich zu machen. Bitte maximal 3 Kategorien nennen. Die Kategorien dienen dem Programmkommittee zur Feststellung thematischer Schwerpunkte der Einreichungen. Sie sind nicht Grundlage für Annahme bzw. Ablehnung eines Beitrags.

Qualitätsaspekte der Wiederverwendung von Schemata

Martin Bertram

debis Systemhaus Dienstleistungen GmbH
Lyoner Straße 9; D-60528 Frankfurt

Zusammenfassung: Ein erheblicher Anteil des Aufwandes zur Erstellung eines Fachkonzepts wird für die Erarbeitung des Datenmodells verwendet. Dieser Aufwand läßt sich durch eine geplante Wiederverwendung von Schemata reduzieren, denn damit kann das gesammelte Modellierungswissen eines Unternehmens gebündelt und wiederverwendet werden. Wesentliche Voraussetzungen für diese Vorgehensweise ist eine hohe Qualität sowohl des Prozesses, als auch der wiederzuverwendenden Schemata selbst. Dieser Beitrag stellt die wesentlichen Qualitätskriterien für diese Schemata vor und verdeutlicht sie anhand eines Beispiels. Abschließend wird die Vorgehensweise insgesamt bewertet.

1 EINLEITUNG

In der heutigen wirtschaftlichen Situation der meisten Unternehmen wächst der Druck zur Reduktion der Kosten der Softwareentwicklung. Insbesondere gilt dies für Beratungsunternehmen, die Software für Kundenunternehmen im Wettbewerb mit anderen Unternehmen erstellen. Ein zentraler Punkt ist hierzu die systematische Wiederverwendung des Wissens, das sich im Laufe vieler Projekte angesammelt hat. Die Wiederverwendung von Wissen geschah bis vor nicht allzu langer Zeit dadurch, daß einzelne Mitarbeiter ihr angehäuftes Wissen in neue Projekte einbrachten, quasi als Wiederverwendung durch Mitarbeitertransfer. Dieses Wissen war aber ausschließlich personenbezogen, mit all den Risiken und Beschränkungen, die sich daraus ergeben. Seit längerem wird nun daran gearbeitet, das Wissen zu systematisieren und personenunabhängig sowie für das gesamte Unternehmen zugänglich zu machen.

Zunächst könnte man daran denken, einfach die einmal erstellten Entwicklungsprodukte direkt wiederzuverwenden. Dies ist aber in der Regel aufgrund

rechtlicher Beschränkungen kaum möglich und es kann, aufgrund der Verschiedenheit der Kundensituationen, nur ein sehr niedriger Wiederverwendungsfaktor erwartet werden. Ein zweiter möglicher Weg ist die Erarbeitung von Referenzmodellen, die das Wissen branchenbezogen bündeln. Dieser Weg erfordert aber vom erstellenden Unternehmen einen erheblichen Aufwand, vor allem da für jedes Anwendungsgebiet ein eigenes, vollständiges und umfassendes Referenzmodell erstellt werden muß¹. Im folgenden wird ein weiterer Weg vorgestellt, nämlich die Wiederverwendung von abstrakter Schemata im Rahmen von Datenmodellen. Solche Schemata werden in einer zentralen Bibliothek gesammelt und den Projekten zugänglich gemacht.

Die semantische Datenmodellierung ist seit langem die verbreitetste Methode, die Anforderungen an Systeme zu beschreiben, konzeptionelle Modelle von Anwendungssystemen zu erstellen sowie auch zur Darstellung unternehmensweiter Datenmodelle. Der Aufwand für diese Modelle stellt einen großen Anteil des Gesamtaufwands eines Fachkonzepts dar und steigt stark mit der Größe der Modelle an. Einer der wesentlichen Gründe dafür liegt in der Tatsache begründet, daß das lehrbuchgemäße Vorgehen eine Modellerstellung auf Basis der Modellierung einzelner Tatbestände - sprich Entities und Relationships - fordert: "Ein Entity ist ein Konzept, eine Person, ein Ding, ...".

Die Wiederverwendung abstrakter Schemata bedeutet eine Möglichkeit, Datenmodelle schneller durch Nutzung größerer komplexer und abstrakter Elemente als Bausteine zu erstellen. Grundlage für diesen Ansatz ist das Vorhandensein einer Bibliothek solcher Elemente. Die Hauptvorteile dieses Ansatzes sind, neben der Verkürzung und Erleichterung der Modellierungsphase, die Möglichkeit zur Aufwandseinsparung sowohl in der Design-, als auch in der Implementierungsphase. Letzteres wird durch die Wiederverwendung vorgefertigter Designs und Implementierungen der einzelnen Schemaelemente erreicht. Als zusätzlicher Effekt wird auch eine spürbare Verbesserung der Qualität des Datenmodells² insgesamt erreicht.

Der Beitrag beschränkt sich auf einen wesentlichen Aspekt der Wiederverwendung abstrakter Schemata, nämlich auf die Beschreibung der für ihre effektive Anwendung notwendigen Qualitätskriterien. Diese sind aus der Erfahrung mit

1 Im debis Systemhaus existieren einige Referenzmodelle, u.a. ICIS für den Versicherungsreich. [dSH94]

2 Bezüglich der Qualität von Datenmodellen, speziell für unternehmensweite Datenmodelle, vergleiche [BER93]

abstrakten Schemata entstanden, lassen sich aber zu großen Teilen auch auf Teile anwendungsspezifischer konzeptioneller Modelle übertragen.

Der erste Teil des Beitrages definiert den Begriff **Schema-Schablone** mit seinen Inhalten. Der Hauptteil beschreibt die auf die Schema-Schablonen anzuwendenden Qualitätskriterien und verdeutlicht sie anhand eines durchgängigen Beispiels. Die Qualitätskriterien setzen sich aus semantischen und konstruktiven Anforderungen³ zusammen. Abschließend werden die Vorteile und die wesentlichen Problembereiche dieses Ansatzes bewertet.

2 EINORDNUNG

Der größte Teil der Diskussionen in der Literatur über das Thema "Wiederverwendung in der Systementwicklung" beschäftigt sich mit der Wiederverwendung des Codes der betreffenden Softwaresysteme⁴. Trotz der hohen Bedeutung, die dieses Gebiet in der Praxis hat, beschäftigen sich nur wenige Arbeiten mit der Wiederverwendung von Dokumenten der frühen Phasen der Softwareentwicklung. Dazu zählen vor allem Konzepte für die Wiederverwendung von Teilen konkreter Datenmodelle⁵. Das reicht hin bis zur Schaffung standardisierter branchenspezifischer Referenzdatenmodelle.

Die Wiederverwendung abstrakter Teilschemata wurde bisher noch kaum untersucht. Das Konzept objektorientierter Patterns⁶ wurde beschrieben und Beispiele für Patterns angegeben. Es wird aber dort weder das Problem der Anwendung in konkreten Situationen noch die Frage der Qualität behandelt.

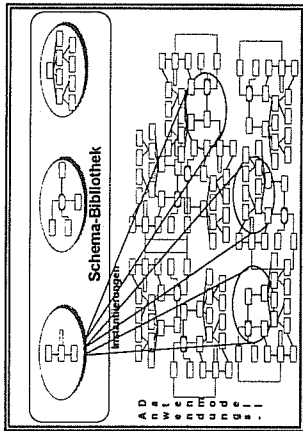


Abbildung 1: Grundidee der Wiederverwendung

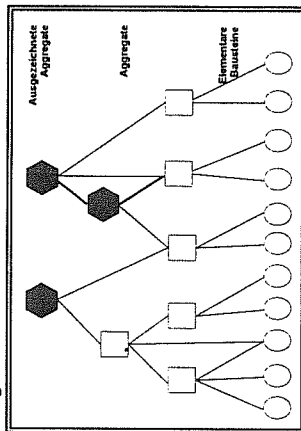


Abbildung 2: Modell der Stückliste

Es ist nicht die Absicht des vorliegenden Beitrages, eine neue Modellierungsmethodik einzuführen, sondern es sollen Qualitätskriterien, speziell für abstrakte Konstrukte semantischer Datenmodelle, angegeben werden, die für den Prozeß der Modellierung großer Anwendungssysteme bis hin zur Unternehmensmodellierung wesentliche Verbesserungsmöglichkeiten bieten.

3 DER BEGRIFF SCHEMA-SCHABLONE

Eine Schema-Schablone ist eine Teilmenge eines abstrakten Datenmodells, das keinen Bezug zu einem konkreten Anwendungsbereich hat, sondern ausschließlich Strukturinformationen darstellt. Die Benennung der Elemente bezieht sich auf die abgebildeten Konzepte. Für die Nutzung in konkreten Datenmodellen muß eine konkrete Instanz des abstrakten Schemas geschaffen werden. Um effizient genutzt werden zu können, müssen Schema-Schablonen die im folgenden beschriebenen Komponenten beinhalten.

Als Beispiel für die im folgenden beschriebenen Qualitätskriterien soll die Stücklistenschablone dienen. Sie dient zur Darstellung mehrstufiger Aggregation von Elementarkomponenten zu einer Einheit.

3.1 SPEZIFIKATION

Die Spezifikation soll die Leistungsfähigkeit des Modells, das Inhalt der Schablone ist, beschreiben. Sie besteht aus den folgenden Bestandteilen:

EIN MODELL

Dabei handelt es sich um eine modellhafte Vorstellung im Sinne einer wohldefinierten, aber nicht notwendigerweise formalen Beschreibung der von der Schablone repräsentierten abstrakten Anwendungskonzepte.

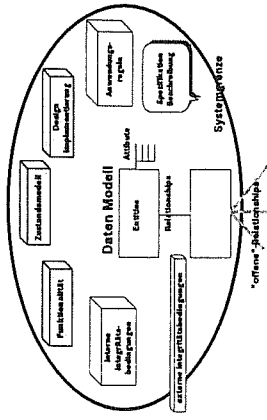


Abbildung 3: Schema-Schablone

³ Auf weitere, d.h. technische und organisatorische Anforderungen wird aus Platzgründen nicht eingegangen.
⁴ Eine Übersicht bietet [KRU92]
⁵ Siehe [CDAZ92]
⁶ Siehe [COAD92]

ANWENDUNGSREGELN

Das sind alle Regeln, die beschreiben, unter welchen Umständen und wie die Schablone innerhalb eines Datenmodells genutzt werden kann und wie nicht. Dazu gehören auch Beispiele für einen sinnvollen Einsatz und solche, in denen die Schablone nicht angewandt werden kann.

VARIANTEN DER BASISSCHABLONE

Varianten dienen dazu, zusätzliche Anwendungsbereiche abzudecken zu können, die von der Basisschablone selbst nicht abgedeckt werden. Diese Anwendungsbereiche unterscheiden sich aber nicht grundsätzlich von denen der Basisschablone, sondern stellen eine Erweiterung bzw. eine Spezialisierung dar. Die Basisschablone sollte in diesem Sinne minimal sein.

Die Beschreibung sollte sich im Sinne einer Vererbung der Eigenschaften auf die Darstellung der Unterschiede zur Basisschablone beschränken.

KLASSIFIKATION

Die Klassifikation der Schablone selbst und ihrer Varianten gemäß einem facetierten Schema dient zur Auffindung der in einer konkreten Situation anwendbaren Schablonen. Dies kann um Synonyme der zugrundeliegenden Fachbegriffe ergänzt werden.

VERWEISE AUF ANDERE SCHABLONEN

Wenn sich ein vorgegebenes Problem weder von der gefundenen Basisschablone, noch von den zugehörigen Varianten lösen lässt, erleichtert ein Verweis auf ähnliche Schablonen der Bibliothek die weitere Suche.

3.2 DATENMODELL

Das Datenmodell einer Schablone beinhaltet die folgenden Bestandteile:

Beispiel:

- Nur die ausgezeichneten Aggregate und die Elementarbausteine können anderen Elementen des Modells zugeordnet werden.
- Jeder Elementarbaustein kann nur genau einem Element zugeordnet werden.

Beispiel:

- In der Basisvariante sind nur Beziehungen der Art "enthält" zwischen Komponenten erlaubt.
- Eine Variante bietet zusätzlich beliebige variable Beziehungen zwischen den Komponenten.

DATENMODELL

Ein semantisches Datenmodell mit graphischer Darstellung, abstrakten Entities und Relationships einschließlich der Definitionen.

EIN SCHNITTSTELLENMODELL

Dieses beschreibt alle Konstrukte des Datenmodells, die die Systemgrenze der Schablone überschreiten. Dies ist die Modellierung der Schnittstellendefinition. Dazu zählen die folgenden Komponenten:

- Eine Beschreibung aller "offenen" Relationships, die die Beziehungen zwischen den Entities der Schablone und denen des Modells herstellen. Hierunter können neben den assoziativen Relationships auch Sub- und Supertyp-Beziehungen fallen.

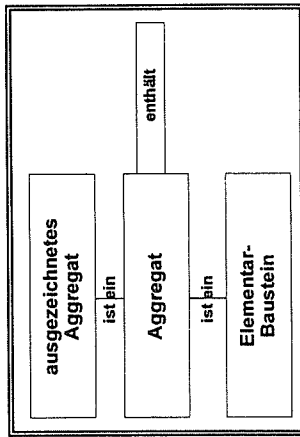


Abbildung 5: Datenmodell der Stückliste

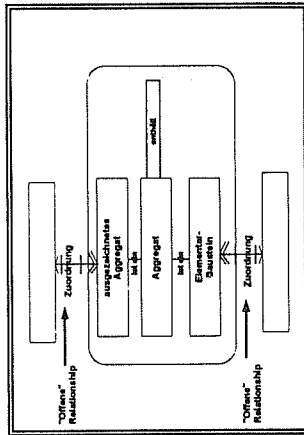


Abbildung 4: Schnittstellen zur Umwelt

- Die Definition aller Integritätsbedingungen, die Elemente sowohl innerhalb als auch außerhalb der Schablone betreffen. Dazu zählen sowohl die impliziten, als auch die expliziten Integritätsbedingungen.

INTEGRITÄTSBEDINGUNGEN INNERHALB DER SCHABLONE

- implizite Integritätsbedingungen, wie z.B. die Bedingungen an die Kardinalitäten.
- explizite Integritätsbedingungen, d.h. solche, die sich nicht mit dem vorgegebenen Formalismus abbilden lassen, sondern die durch eigene Konstrukte, wie z.B. Prädikatenlogik, formuliert werden müssen.

ATTRIBUTE

- Definition und Beschreibung der abstrakten Attribute mit den zugehörigen Datentypen.

3.3 OPTIONALE KOMponentEN⁷

Um den Nutzen der Schablonen zu maximieren, sind über die oben beschriebenen Komponenten hinaus die folgenden Bestandteile sinnvoll:

FUNKTIONALITÄT

Zur Erleichterung der Funktionsmodellierung sollten zu einer Schablone auch die anwendbaren Funktionen angegeben werden. Die vorhandenen Varianten spiegeln sich auch in den Funktionen wieder.

ZUSTANDSMODELL

Zur Beschreibung der Lebensläufe der Elemente dient das Zustandsmodell für das der Schablone zugrundeliegende Konzept (falls anwendbar), sowie gegebenenfalls für die Teilkonzepte einschließlich der definierten Varianten.

DESIGNKONZEPTE UND IMPLEMENTIERUNG

Zur Erleichterung des Systemdesigns sollten zu einer Schablone auch die möglichen Designkonzepte mit ihren Stärken und Schwächen angegeben werden. Diese können für unterschiedliche Varianten verschieden sein. Aufbauend auf diesen Designkonzepten ist es sinnvoll, vorgedachte Implementierungen anzubieten.

3.4 BIBLIOTHEK

Zentrales Werkzeug der Wiederverwendung von Schemata ist die Bibliothek, die die Schablonen enthält und die Infrastruktur zur Wiederverwendung zur Verfügung stellt. Die Qualität der Schema-Schablonen bestimmt nicht nur die

Effizienz und Qualität aller Prozesse zur Wiederverwendung von Schemata, sondern auch die Qualität des Modells der Anwendung als Ganzes.

Qualität in diesem Zusammenhang bedeutet vor allem Qualität der Infrastruktur, d.h. sowohl die Ergonomie der Benutzeroberfläche, als auch der Umfang der angebotenen Funktionalität.

4 QUALITÄTSKRITERIEN

Die Qualität einer Schema-Schablone kann durch eine Menge an Anforderungen beschrieben werden; die alle diejenigen Charakteristiken betreffen, die sicherstellen, daß die Wiederverwendung eines Schemas möglich und effizient bzw. effektiv ist.

Die Anforderungen lassen sich teilweise aus den Anforderungen an wiederverwendbare Softwarekomponenten⁸ bzw. aus den Kriterien einer guten Softwarearchitektur⁹ herleiten. Sie lassen sich in Anforderungen an die Semantik und an die Konstruktion der Schablone gliedern.

Als Beispiel für die folgenden Qualitätskriterien soll ebenfalls die Schablone "Stückliste" dienen.

4.1 QUALITÄTSKRITERIEN BEZÜGLICH DER SEMANTIK

WOHLDEFINIERT ANWENDUNGSSEMANTIK

Die Schema-Schablone muß ein Modell eines eindeutig definierten Ausschnittes des Betrachtungsgegenstandes sein, d.h. es muß eine eindeutig definierten Menge von Anwendungsfakten repräsentieren. Dieser Ausschnitt sollte weder zu groß - und damit zu schwer anwendbar -, noch zu klein bzw. trivial, sein um nützliche Bausteine abzugeben.

Das in Abbildung 2 dargestellte Konzept einer Stückliste stellt ein klar abgegrenztes fachliches Konzept dar, das einen angemessenen Umfang hat.

PRÄZISE SPEZIFIKATION

Spezifikation bedeutet die Beschreibung dessen, **was** die Schema-Schablone repräsentiert, nicht **wie** sie das tut. Übertragen auf die Schablonen bedeutet dies, daß es nicht nötig sein darf, das interne Datenmodell einer Schablone zu

| |
|---|
| Typische Funktionen der Stückliste sind: |
| • Einfügen und Löschen von Komponenten. |
| • "Explosion" eines Aggregats, d.h. Liste aller elementaren Bestandteile. |
| • Verwendungsnachweis einer Elementarkomponente. |

⁷ Auf die Qualitätsaspekte der optionalen Komponenten wird im folgenden nicht näher eingegangen.

⁸ Einen Überblick enthält [KKR92].

⁹ Modularisierungskriterien sind beschrieben bei [PAR72a].

hinaus der Aufwand der Anpassung an ein spezifisches Anwendungsmodell zu groß. Durch eine hohe Universalität in der Anwendung wird die Anzahl der zur Verfügung zu stellenden Schablonen klein gehalten und damit sowohl der Verwaltungsaufwand der Bibliothek selbst, als auch der Suchaufwand innerhalb der Bibliothek verringert.

Das Konzept der Stückliste läßt sich nicht nur auf technische Gegenstände anwenden, sondern auch auf die Zusammenfassung von Firmen zu Konzernen oder auf die Gliederung von Verträgen in Klauseln und Paragraphen.

4.2 QUALITÄTSKRITERIEN BEZÜGLICH DER MODELLIERUNG BZW. KONSTRUKTION

MODELLIERBARKEIT

Die Schema-Schablone muß in dem Formalismus modellierbar sein, der für das Modell der Anwendung verwendet wird. Mindestens aber muß eine Abbildung von dem in der Bibliothek verwendeten Formalismus in den Formalismus der Anwendungsmodelle gegeben sein. Das Datenmodell der Schablone selbst muß den etablierten Qualitätskriterien eines Datenmodells genügen. Die Schablone muß vor allem in sich konsistent sein. Es ist offensichtlich, daß eine Verletzung dieses Kriteriums die Anwendbarkeit in konkreten Situationen stark erschwert, da dann eine Transformation des Formalismus nötig wird.

ORTHOGONALITÄT

Die einzelnen Schema-Schablonen müssen so konstruiert sein, daß möglichst viele Kombinationen zwischen ihnen möglich sind. Solche Kombinationen dienen zur Abbildung noch komplexerer Anwendungsprobleme. Auf diese Art kann die Menge der Schema-Schablonen klein gehalten werden, da sich die Anwendungsfälle in der Regel durch solche Kombinationen abdecken lassen. Konsequenz daraus ist, daß jedes abstrakte Konzept nur durch genau eine Schablone abgebildet werden darf, da andernfalls überschneidende Konzepte Widersprüche erzeugen. Jede Schablone muß die Stellen, an denen andere Schablonen eingesetzt werden können, sowie die Minimalanforderungen an die einzusetzende Schablone präzise definieren. Die Beschreibung einer Schablone darf natürlich umgekehrt keinerlei Verweise darauf enthalten, wo sie selbst eingesetzt werden kann. Man kann sich z.B. neben der Schablone "Stückliste" eine Schablone "Historisierung" vorstellen. Diese beiden Schablonen sind voneinander vollständig unabhängig. Sie können aber zu einer gemeinsamen Schablone "historisierbare Stückliste" kombiniert werden. Andererseits könnte

kennen, um zu verstehen, wie und wofür sie angewandt werden kann. Eine solche Spezifikation darf nicht nur für die Systemanalytiker, sondern muß auch für die Anwender und Auftraggeber verständlich sein, die die Korrektheit und Konsistenz des Modells zu beurteilen haben.

Die Spezifikation enthält die wesentlichen fachlichen Konzepte und ist auch ohne den Bezug auf das Datenmodell für Fachbereichsmitarbeiter verständlich.

Eine präzise Spezifikation verringert den Dokumentationsaufwand für das konkrete Modell, da sie nur noch in die aktuelle Umgebung übertragen werden muß. Darüber hinaus wird die Qualität der Dokumentation des gesamten Datenmodells als Bestandteil des Fachkonzepts verbessert. Es werden nicht nur einzelne Elemente der Modelle beschrieben, sondern größere Zusammenhänge, die es wesentlich erleichtern, das Modell als Ganzes zu verstehen.

MINIMALE KOGNITIVE DISTANZ

Es muß weniger aufwendig sein, eine Schablone zu finden, zu verstehen und anzuwenden, als es ist, den abgebildeten Teil der Anwendung vollständig neu zu modellieren.

Konkret bedeutet das, daß es einfach sein muß, zu den meist unscharf formulierten, fachlichen Anforderungen die passende Schema-Schablone zu finden. Auch dazu dient in erster Linie das Modell des Konzepts und nicht sein Datenmodell, sowie eine präzise Spezifikation mit den Anwendungsregeln.

PRÄZISE NUTZUNGSREGELN

Es sind präzise explizite Regeln nötig, die eine Identifizierung derjenigen Ausschnitte der Anwendungswelt erlauben, für deren Modellierung die Schema-Schablone angewandt werden kann. Darüber hinaus sind Regeln für die Anwendung der Schablone selbst sowie für die Grenzen der Anwendbarkeit nötig.

HINREICHENDE UNIVERSALITÄT

Eine Schema-Schablone muß hinreichend abstrakt sein, um in möglichst vielen Anwendungsmodellen benutzt werden zu können. Andererseits darf sie nicht zu abstrakt sein, da dadurch die kognitive Distanz zu groß werden kann. Bei sehr abstrakten Schablonen wird darüber

Beispiel:
Die Nutzungsregeln im Falle der Stückliste könnten beinhalten:

- "Die Stückliste sollte bei allen Formen mehrstufiger Aggregation elementarer Bausteine zu einem Aggregat angewandt werden...."
- "Nicht angewandt werden sollte die Stückliste bei einfachen Gruppenbildungen...."

die Schablone "Historisierung" mit einer Schablone "mehrdimensionale Tabelle" zu einer Schablone "historisierbare mehrdimensionale Tabelle" kombiniert werden.

KOMPAKTHEIT

Ein Kriterium für den Umfang einer Schema-Schablone ist die Tatsache, daß in einer Schablone nur solche Konzepte zusammengefaßt sind, die im Sinne einer funktionalen bzw. informalen Kopplung¹⁰ zusammengehören. Dabei handelt es sich um eine weitergehende Bedingung als es die Orthogonalität ist, denn dort wird nur die Überschneidungsfreiheit gefordert. Als Basiskriterium dient die Definition auf einer gemeinsamen Datenmenge. Übertragen auf die Schablone bedeutet dies, daß als notwendige Bedingung das Datenmodell einer Schablone zusammenhängend sein muß, entweder mittels Beziehungen, gemeinsamen Regeln oder durch Integritätsbedingungen. Dadurch wird es unnötig, nur Teilmengen der Schema-Schablonen anzuwenden, die innerhalb dieser Schablonen unter Umständen nur schwer abgrenzbar sind.

VOLLSTÄNDIGKEIT

Die Schema-Schablone muß genau alle Aussagen über das abgebildete Konzept enthalten, die korrekt und relevant sind. Das beinhaltet die Geschäftsregeln, die die Integrität und die Konsistenz des modellierten Teils des Anwendungsgebiets beschreiben. Diese Regeln definieren sowohl die Einbettung in das Anwendungsmodell, als auch die internen Bedingungen des Konzepts.

Es existieren Regeln, die unabhängig von der konkreten Anwendung sinnvoll sind. Im Einzelfall kann es auch ausreichen, klar festzulegen, welche Regelungen im konkreten Anwendungsfall noch zu spezifizieren sind.

HOHE FLEXIBILITÄT

Die Schema-Schablone sollte so flexibel sein, daß möglichst viele Veränderungen in der modellierten Anwendungswelt ohne Veränderung oder Austausch der Schablone abgebildet werden können. Solche Flexibilität könnte durch die Definition von Regeln auf der Basis von Datenstrukturen erreicht werden. Auf diese Art können die Regeln teilweise auch von der Mitarbeitern der Fachbereiche gepflegt werden. Solche Datenstrukturen können ebenfalls als Schema-Schablonen aufgefaßt werden.

Es sollte zunächst versucht werden, Änderungen in der Anwendungswelt innerhalb der verwendeten Schablone abzubilden. Je weiter diese Flexibilität geht, desto später wird es nötig, von einer Schablone auf eine ihrer Varianten überzugehen, oder gar den Wechsel zu einer anderen Schablone vornehmen zu müssen. Eine hohe Flexibilität verringert den Prüfungsaufwand, da nicht im kleinsten Detail analysiert werden muß, um festzustellen, ob eine Variante einer Schablone anwendbar ist.

WOHLDEFINIERTE SCHMALE SCHNITTSTELLE

Eine Schema-Schablone muß ein wohldefinierte Schnittstelle - im oben definierten Sinn - zur Umgebung der Schablone innerhalb des Datenmodells besitzen. Die Breite dieser Schnittstelle sollte minimal sein, d.h. so wenige Relationships und Regeln wie sinnvollerweise möglich, enthalten.

Basierend auf einer wohldefinierten Schnittstelle ist es möglich, eine Schablone durch eine andere mit derselben oder einer kompatiblen Schnittstelle auszutauschen, da die Anzahl der herzustellenden bzw. zu ändernden Verbindungen zwischen der Schablone und dem übrigen Modell gering ist. Ein solcher Austausch, der notwendig werden könnte, wenn Änderungen in der Anwendung die eingebaute Flexibilität der benutzten Schablone übersteigen, wird durch eine schmale Schnittstelle somit wesentlich erleichtert.

VORHANDENSEIN VON VARIANTEN

Es erscheint nicht sinnvoll, jedes Problem sofort mit einem sehr mächtigen Modell zu beschreiben, da dieses Möglichkeiten bietet, die in vielen Fällen nicht genutzt werden können. Durch den Einsatz einer zu mächtigen Schablone erhöht sich der Aufwand sowohl in der Modellierungsphase als auch bei der späteren Implementierung.

Um die Zahl der Modellierungskonstrukte insgesamt zu beschränken, sollte die Zahl der Varianten ihrerseits beschränkt bleiben. Varianten sollten nur bei substantiellen Unterschieden beschaffen werden, die nicht durch die eingebaute Flexibilität behandelt werden können. Die Suche innerhalb der Varianten einer Schema-Schablone ist, selbst in einer gut organisierten Bibliothek, weniger aufwendig, als die Suche in einer Menge unabhängiger Schablonen, da hier immer nur die Unterschiede zur Basisschablone beschrieben und somit zu beurteilen sind.

¹⁰ Eine Definition enthält [BAL82].

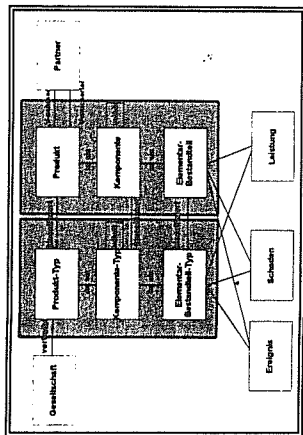


Abbildung 6: Basisvariante

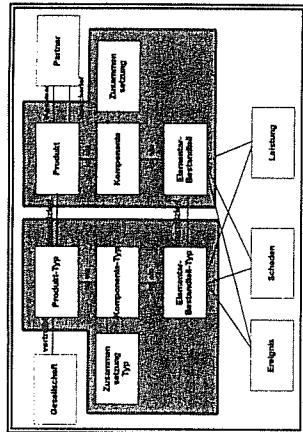


Abbildung 7: Variante mit variablen Beziehungen

Die Basisvariante der Stückliste umfaßt nur die Beziehung "enthält" zwischen den Komponenten, die Variante läßt beliebige flexible und datenmäßig definierbare Beziehungen zwischen den Komponenten zu.

Die Menge der Varianten sollte halbgeordnet sein mit dem Hintergrund einer Aufwärtskompatibilität zwischen den Varianten. Dadurch wird es einfach möglich, eine Variante durch eine leistungsfähigere zu ersetzen, ohne daß das überrige Modell deswegen angepaßt werden muß.

Aufwärtskompatibilität bedeutet hier konkret, daß die leistungsfähigere Variante mindestens das leisten muß, was die Basisvariante kann. Dabei bezieht sich der Begriff "leisten" auf den Umfang der fachlichen Gegebenheiten, die abgebildet werden. Insbesondere müssen mindestens die Relationships und Integrationsregeln definiert sein, die in der Basisvariante vorhanden sind.

5 ZUSAMMENFASSUNG UND BEWERTUNG

Im vorliegenden Beitrag wurden die Qualitätskriterien für die Wiederverwendung von abstrakten Teilschemata semantischer Datenmodelle beschrieben und ihr Einfluß auf die Modellierungsprozesse dargestellt. Die zugrundegelegte Vorgehensweise basiert auf einer Bibliothek abstrakter Schema-Schablonen. Die angegebenen Qualitätskriterien bestimmen ganz wesentlich die Effizienz und Effektivität des Modellierungsprozesses.

Obwohl noch keine harten quantitativen Daten bezüglich der erzielten oder erzielbaren Einsparungen in den für unser Unternehmen typischen Projekten vorliegen, lassen sich doch auf der vorhandenen Basis folgende qualitative Aussagen zu den Vor- und Nachteilen der vorgestellten Vorgehensweise machen:

✚ Die Qualität der entstandenen Datenmodelle hat sich erhöht. Dabei wird die Schablone oft als "Liste der zu stellenden Fragen" benutzt und damit die vollständige Bearbeitung des Problembereichs sichergestellt.

✚ Da die Konzepte der Schablonen als in sich geschlossenes Konzept erläutert werden können, erzielt man eine Verbesserung der Verständlichkeit durch Einführung einer höheren Abstraktionsebene oberhalb der Elementarkonstrukte der Datenmodellierung.

- Der Aufbau und die Pflege der verwendeten Bibliothek sind aufwendig. Dabei wurde vor allem der Aufwand, eine Schemaschablone gemäß den vorgestellten Qualitätskriterien zu beschreiben, zunächst unterschätzt.

- Die Akzeptanz dieser Vorgehensweise litt unter dem "Not Invented Here"-Syndrom, dadurch daß wesentliche Inhalte von außen vorgegeben und nicht im Projekt selbst entwickelt wurden. Die Vorteile sind aber offensichtlich und man kann deshalb von einer wachsenden Akzeptanz ausgehen.

Zusammenfassend läßt sich sagen, daß die Vorteile dieser Vorgehensweise der Modellierung die Nachteile und Probleme bei weitem überwiegen und ein erhebliches Potential für Einsparungen und Qualitätsverbesserung zu erwarten ist. Durch Beachtung der beschriebenen Qualitätskriterien kann der Nutzen der Schemabibliothek vergrößert und der Erfolg dieser Investition sichergestellt werden. Für Beratungsunternehmen wird ein solcher Fundus von Schema-Schablonen zukünftig eine hohe Bedeutung für die Wettbewerbsfähigkeit haben.

6 LITERATUR

- BAL82 Balzert H., Die Entwicklung von Software-Systemen; Bibliographisches Institut 1982.
 BER93 Bertram M., Aspekte der Qualitätssicherung von Unternehmensdatenmodellen; Proceedings "Wirtschaftsinformatik '93 Münster", pp 230-242
 CDAZ92 Castano S., DeAntonellis V., Zonta B., Classifying and Reusing Conceptual Schema; Proceedings "11th ER-Conference 1992", pp.121-138.
 COAD92 Coad P., Object Oriented Patterns; Communications of the acm Vol.35(9) 1992, pp.152-159.
 dSH94 ICIS Insurance Company Information System Leistungsübersicht; debis Systemhaus Dienstleistungen GmbH 1994.
 KRU92 Krueger C.W. Software Reuse. acm Computing Surveys; Vol.24 No.2 (1992), pp.131-184.
 PAR72 Parnas D.L. On the Criteria to be used in Decomposing Systems into Modules; Communications of the acm; Vol.15 No.12 (1972), pp. 1053-1058.

x Jan

Entwicklung zuverlässiger Informationssysteme*

S. Conrad¹, G. Denker², M. Gogolla³, R. Herzig³, N. Vlachantonis⁴ und H.-D. Ehrich²

¹ Universität Magdeburg, Institut für Technische Informationssysteme,
Postfach 4120, D-39016 Magdeburg

² Technische Universität Braunschweig, Informatik, Abt. Datenbanken,
Postfach 3329, D-38023 Braunschweig

³ Universität Bremen, FB 3 Mathematik und Informatik,
Postfach 330440, D-28334 Bremen

⁴ LION, Gesellschaft für Systementwicklung mbH,
Universitätsstr. 140, D-44799 Bochum

1 Einleitung

Die zunehmende Komplexität von Softwaresystemen führt dazu, daß Fragen der Zuverlässigkeit und Korrektheit solcher Systeme eine immer zentralere Bedeutung erlangen. Ein Beleg hierfür sind die verschiedenen nationalen und internationalen Bemühungen um Zertifikationsverfahren für Software [Krü93, Rom93]. Insbesondere unter wirtschaftlichen Gesichtspunkten erweist es sich als sinnvoll, schon in den frühen Phasen geeignete Gütekriterien zu berücksichtigen, um späteren, meist kostenintensiven Wartungs- und Anpassungsaufwand in engen, kalkulierbaren Grenzen halten zu können.

Darüber hinaus zeigen verschiedene Bemühungen um eine Standardisierung des Softwareentwurfsprozesses (z.B. das V-Modell [BD93]), daß in Zukunft auch die Anforderungen der Auftraggeber den Nachweis einer durchgängigen Entwicklungsmethodik und den Einsatz formaler Methoden notwendig machen werden. Insbesondere in Hinblick auf sicherheitskritische Eigenschaften wird bald deren formaler Nachweis Vertragsgrundlage werden.

Diese sich abzeichnende Entwicklung hat uns dazu veranlaßt, die formalen Aspekte bei der Entwicklung von Informationssystemen (als eine große Klasse von Softwaresystemen) näher zu untersuchen.

Dies geschah im wesentlichen im Rahmen des Forschungsprojektes KORSo[BJ93], in dem die Konsolidierung formaler Methoden im Mittelpunkt stand. Verschiedene Gruppen arbeiteten dabei zusammen an folgenden Schwerpunkten: Methodik, Beschreibung und Modellierung, Fallstudien und Werkzeuge.

Die Arbeiten unserer Gruppe dabei lassen sich wie folgt zusammenfassen: Ausgehend von einem einfachen objektorientierten Ansatz zur Modellierung von Objektgesellschaften als Informationssysteme wurden insbesondere eine geeignete Entwicklungsumgebung konzipiert, der Bereich Validation von Objektspezifikationen mittels Animation untersucht und ein Beweisunterstützungssystem entworfen, das formale Nachweise von Objekteigenschaften ermöglicht. Ziel ist die Entwicklung von Informationssystemen auf der Grundlage von objektorientierten Datenbanksystemen [KM93, Pis93].

Im folgenden werden wir zunächst die Objektbeschreibungssprache TROLL *light* vorstellen, die als Grundlage der in Angriff genommenen Aufgaben essentielle Konzepte zur Informationssystem-Modellierung mittels interagierender Objekte bietet. Dann geben wir eine kurze Übersicht über die zugehörige Entwicklungsumgebung bevor wir näher auf zwei wesentliche Komponenten, den Animator und das Beweisunterstützungssystem, eingehen.

*Das hier präsentierte Projekt wurde als Teil des BMFT-Verbundprojektes KORSo (= Korrekte Software) unter der Förd.Nr. 01 IS 203 D in Braunschweig durchgeführt.

2 Konzepte von TROLL *light*

TROLL *light* [CGH92, GCH93, HCG94] ist eine Sprache zur Beschreibung von strukturellen und dynamischen Eigenschaften von Objekten. Die Beschreibung von strukturellen Eigenschaften orientiert sich in weiten Teilen an semantischen Datenmodellen.

- Die in einem Zustand beobachtbaren Eigenschaften eines Objektes werden durch Attribute beschrieben. Einfache Attribute sind daten- oder objektwertig. Darüber hinaus erlauben vordefinierte Sortenkonstruktoren wie *set* oder *tuple* auch die Spezifikation komplexer Attributbereiche.
- TROLL *light*-Objekte sind in Objekthierarchien, die sich aus Unterobjektbeziehungen ergeben, organisiert. Eine Unterobjektbeziehung ist hierbei als eine exklusive „Teil von“- bzw. Komponentenbeziehung zu verstehen.
- Normalerweise werden Attributwerte direkt durch das Eintreten gewisser Ereignisse bestimmt. Daneben ist es auch möglich, abgeleitete Attribute zu spezifizieren, d.h. Attribute, deren Inhalte sich aus anderer gespeicherter oder abgeleiteter Information bestimmen. Zur Formulierung von Ableitungsregeln stellt TROLL *light* einen SQL-ähnlichen Anfragekalkül zur Verfügung.
- Der Anfragekalkül von TROLL *light* unterstützt auch die Spezifikation statischer Integritätsbedingungen.

Die Beschreibung von dynamischen Eigenschaften basiert auf der Spezifikation von Ereignissen. Ereignisse sind Abstraktionen von zustandsverändernden Operationen auf Objekten.

- Objekt Ereignisse werden durch eine endliche Menge von Ereignisgeneratoren beschrieben. Jeder Ereignisgenerator kann mit einer Liste von Parametersorten versehen sein. Ein Ereignisgenerator mit aktuellen Parameterwerten liefert ein Ereignis.
- Die Auswirkung von Ereignissen auf Attribute werden durch Auswertungsregeln beschrieben.
- Ereignisse in verschiedenen Objekten können

durch Interaktionsregeln synchronisiert werden.

- Die möglichen Ereignisfolgen können mit Hilfe CSP-ähnlicher Prozeßbeschreibungen auf zulässige Folgen eingeschränkt werden.

Den Rahmen zur Beschreibung all dieser Objekteigenschaften bilden *Templates*. Als Beispiel geben wir in Abb. 1 die Beschreibung von Objekten an, die Autoren in einem Bibliotheksinformationssystem repräsentieren sollen.

Die so beschriebenen Autoren-Objekte haben einen Namen, ein Geburtsdatum und Anzahlen verkaufte Bücher als Attribute, wobei für jedes Jahr eine Anzahl verkaufter Bücher gespeichert werden kann. Ereignisse, die im Leben eines Autoren-Objektes eintreten können, sind sein Geburtseignis, eine Namensänderung, das Abspeichern von Verkaufszahlen und schließlich ein Todesereignis. Im VALUATION Abschnitt ist beschrieben, welche Auswirkungen das Eintreten eines Ereignisses auf Attribute hat. Der Verhaltensteil legt die zulässigen Lebensläufe von Autoren-Objekte fest.

3 Die TROLL *light*-Entwicklungsumgebung

Die Entwicklung großer Software- und Informationssysteme ist ein aufwendiger Prozeß, der verschiedene Methoden und Techniken zur Verminderung des Zeitbedarfs und der Kosten erfordert. Um das zu unterstützen, sind Software-Werkzeuge (Tools) notwendig (z.B. CASE-Werkzeuge wie in [KEK⁺94]). Um effektiv arbeiten zu können, sollten diese Werkzeuge innerhalb einer Software-Entwicklungsumgebung integriert werden. Kern unserer Entwicklungsumgebung ist die zuvor skizzierte Sprache TROLL *light*.

Die TROLL *light*-Entwicklungsumgebung ist ein offenes System, das von seinen Anwendern an neue Anforderungen angepaßt werden kann und in das neue Tools integriert werden können [VHG⁺93]. Bedingungen für die Integration betreffen u.a. die Anpassung neuer Tools, d.h. diese müssen in der Lage sein, von vorhandenen Tools erzeugte Daten und Dokumente (Spezifikationen, Quelltexte, etc.) weiterzuverarbeiten. Das ist ein sehr wichtiger Aspekt

```

TEMPLATE Author
  DATA TYPES  String, Date, Nat;
  ATTRIBUTES  Name:string; DateOfBirth:date;
              SoldBooks(Year:nat):nat;
  EVENTS      BIRTH create(Name:string, DateOfBirth:date);
              changeName(NewName:string);
              storeSoldBooks(Year:nat, Number:nat);
              DEATH destroy;
  VALUATION   [create(N,D)] Name=N, DateOfBirth=D;
              [changeName(N)] Name=N;
              [storeSoldBooks(Y,NR)] SoldBooks(Y)=NR;
  BEHAVIOR    PROCESS AuthorLife1 =
              ( storeSoldBooks -> AuthorLife1 |
                changeName -> AuthorLife2 |
                destroy );
              PROCESS AuthorLife2 =
              ( storeSoldBooks -> AuthorLife2 |
                destroy );
              ( create -> AuthorLife1 );
END TEMPLATE;

```

Abbildung 1: Objektbeschreibung für Autoren.

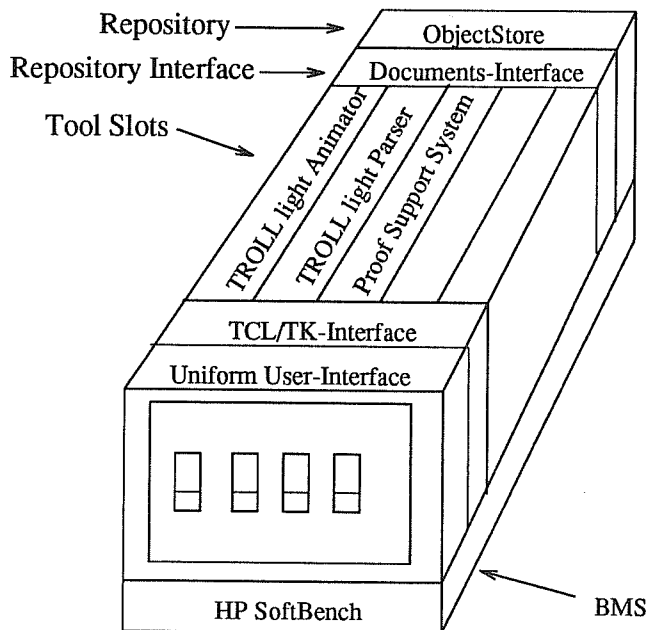


Abbildung 2: Architektur der TROLL light-Entwicklungsumgebung.

da Daten und Dokumente strukturiert in einem Datenbanksystem (Repository) abgespeichert werden. Um die Entwicklungsumgebung weitestgehend offen für Erweiterungen und Anpassungen zu halten, müssen die Tools möglichst unabhängig voneinander sein. Daher unterstützt die TROLL light-Entwicklungsumgebung eine lose Kopplung der Tools. Das wird durch eine Tool-Kommunikation erreicht, die auf dem Austausch von Nachrichten (im Prototyp realisiert durch die HP Softbench) und auf der Speicherung der Daten in einem zentralen Repository basiert [DHS⁺91].

Nachrichten sind hierbei kein Mittel, um Daten auszutauschen, sondern dienen dem Austausch integritätserhaltender Informationen und der Steuerung der Tools [VH93]. Verändert beispielsweise ein Tool ein Dokument im Repository, so verschickt es eine Nachricht darüber an die Umgebung, um andere Tools darüber zu informieren. Der Austausch von Nachrichten wird durch den *Broadcast Message Server* (BMS) verwaltet, der Nachrichten selektiv nur an betroffene Tools weiterleitet. Neue Tools können integriert werden, ohne daß existierende angepaßt werden müssen. Im allgemeinen ist nur eine Aktualisierung der Informationen des BMS über die Wei-

terleitung von Nachrichten nötig.

Ferner ist die TROLL *light*-Entwicklungsumgebung eine Instantiierung des ECMA-Referenzmodells für Entwicklungsumgebungen (siehe Abb. 2). Sie ist auf den dargestellten Integrationsrahmen BMS, ObjectStore [LLOW91] und TCL/TK (User-Interface-Manager) [Ous93] aufgebaut.

Zwei wichtige Komponenten der TROLL *light*-Entwicklungsumgebung sind der Animator und das Beweisunterstützungssystem. Der Animator dient dem Prototyping einer mit TROLL *light* spezifizierten Objektgesellschaft und das Beweisunterstützungssystem hilft beim formalen Nachweis bestimmter Eigenschaften spezifizierter Objekte. Eine andere geplante, hier nicht näher vorgestellte Komponente ist ein Tool, das den Objektansatz von TROLL *light* mit der etablierten Entity-Relationship-Modellierungstechnik verbindet [GHC⁺94].

4 Der Animator

Motivation. Zwischen der angestrebten Formalität und der Verständlichkeit eines konzeptionellen Schemas besteht oftmals ein Zielkonflikt. Man vergleiche zum Beispiel Datenflußdiagramme als intuitive Darstellungen mit vager Semantik („Was ist ein Daten- bzw. Kontroll-, Informations-, Materialfluß?“) mit algebraischen Spezifikationen als Beschreibungen mit klar definierter Semantik, aber aus Benutzersicht deutlich verminderter Zugänglichkeit. Ein formal beschriebenes konzeptionelles Schema muß deshalb zur Validierung dem Auftraggeber oder späteren Anwender eines zu entwickelnden Systems in irgendeiner Weise *vermittelt* werden. Hierzu gibt es verschiedene Techniken.

Eine weitverbreitete Vorgehensweise besteht in der *Umwandlung* eines konzeptionellen Schemas in leichter verständliche Darstellungen. Hierbei kommt insbesondere die Anwendung graphischer Darstellungsmittel oder die Rückübersetzung einer Spezifikation in eine natürlichsprachliche Form in Frage. Ein anderer Ansatz basiert auf der formalen *Analyse* konzeptioneller Schemata. Durch logisches Schließen können aus einem gegebenen Schema bestimmte Folgerungen abgeleitet und mit dem Anwender diskutiert werden. Ein Weg, der in besonderer Wei-

se dazu geeignet ist, eine Modellbildung mit informellen Benutzervorstellungen abzugleichen, besteht im *experimentellen Prototyping* wichtiger Schemabestandteile [LL93].

Das Prototyping funktionaler Spezifikationen erfolgt in der Regel durch die Anwendung spezifizierter Funktionen auf konkrete oder symbolische Eingabewerte. Beim Testen nichtdeterministischer Systeme müssen dahingegen im allgemeinen bestimmte Ereignisse vorgegeben werden. Dies kann unter Umständen durch die Vorgabe gewisser Auftretenswahrscheinlichkeiten simuliert werden. Überläßt man dem Anwender die Vorgabe von Ereignissen zum Austesten bestimmter Szenarien spricht man auch von *Animation* [CRB93].

Animation von Objektbeschreibungen. Die Animation von Objektbeschreibungen durch Vorgabe von Mengen parallel eintretender Ereignisse kann dazu dienen, das spezifizierte Verhalten konzeptioneller Objekte mit dem beabsichtigten Verhalten zu vergleichen. Dabei ist zu beachten, daß die Validierung einer Spezifikation den gleichen Einschränkungen unterliegt wie das Testen eines Programms. Die Feststellung, daß das spezifizierte Verhalten in den getesteten Fällen dem geforderten Verhalten entspricht, kann lediglich dazu dienen, das Vertrauen in eine Spezifikation zu erhöhen. Solange es aber nicht getestete Szenarios gibt, welche das Gegenteil zeigen könnten, bleibt die Korrektheit einer Spezifikation ungewiß.

Architektur des Animationssystems. Die Animation von Objektbeschreibungen sollte innerhalb einer Entwicklungsumgebung softwaremäßig unterstützt werden. Ein entsprechendes Werkzeug muß zum einen die Untersuchung aktueller Zustände einer Objektgesellschaft ermöglichen, zum anderen das Einleiten von Zustandsübergängen durch die Spezifikation von Mengen gleichzeitig eintretender Ereignisse erlauben.

Die Struktur des TROLL *light*-Animationssystems ist in Abb. 3 dargestellt (Pfeile drücken „benutzt“ Beziehungen aus). Das System basiert zunächst auf einem Speicher zur strukturierten Ablage von Objektbeschreibungen (engl. *template dictionary*). Dieser Speicher ist nicht exklusiver Bestandteil des Animationssystems, sondern zentrale Komponente der ganzen Entwicklungsumgebung, welche

zum Beispiel auch vom Parser und dem Beweisunterstützungssystem benutzt wird. Die zweite grundlegende Komponente des Animationssystems stellt eine Objektbank dar, welche als persistenter Speicher für Objektzustände dient.

TROLL *light*-Objektbeschreibungen umfassen Terme (und Formeln) eines SQL-ähnlichen Anfrageskalküls. Zur Auswertung dieser Terme ist ein eigenes Softwaremodul zuständig. Die Auswertung von Termen hängt generell vom aktuellen Zustand einer Objektgesellschaft ab, weshalb der Termauswerter Zugriff auf die Objektbank haben muß. Das Ausführungsmodul stellt das Herz des Animationssystems dar: Zu einem aktuellen Zustand und einer gegebenen Ereignismenge bestimmt das Ausführungsmodul einen möglichen Folgezustand bzw. berichtet über eventuell auftretende Fehlersituationen. Schließlich bildet die Benutzeroberfläche die Schnittstelle zu Anwendern des Animationssystems.

Die Benutzeroberfläche wurde mit Hilfe von TCL/TK realisiert. Objektzustände werden im Rahmen von Fenstern zu Objekten visualisiert, welche unter anderem eine Darstellung der aktuellen Unterobjekte und Attributwerte eines Objekts umfassen (s. Abb. 4). Zusätzlich enthalten Objektfenster eine Liste der spezifizierten Ereignisgeneratoren. Durch Mausklick und Eingabe eventuell verlangter Ereignisparameter können zur Ausführung anstehende Ereignisse in eine spezielle *event collection box* übertragen werden, von wo aus sie schließlich dem Ausführungsmodul zur Berechnung eines möglichen Zustandsübergangs übergeben werden.

5 Das Beweisunterstützungssystem

Motivation. Mit Hilfe der Beweisunterstützungssystem soll es möglich sein, Eigenschaften von Objekten zu formulieren und diese dann unter Verwendung ihrer Spezifikation zu beweisen. Benötigt wird dazu ein Kalkül, in dem solche Eigenschaften ausgedrückt werden können. Ferner müssen TROLL *light*-Spezifikationen in diesen Kalkül übersetzt werden können [Con94]. Der für diese Zwecke entwickelte Verifikationskalkül berücksichtigt alle Konzepte

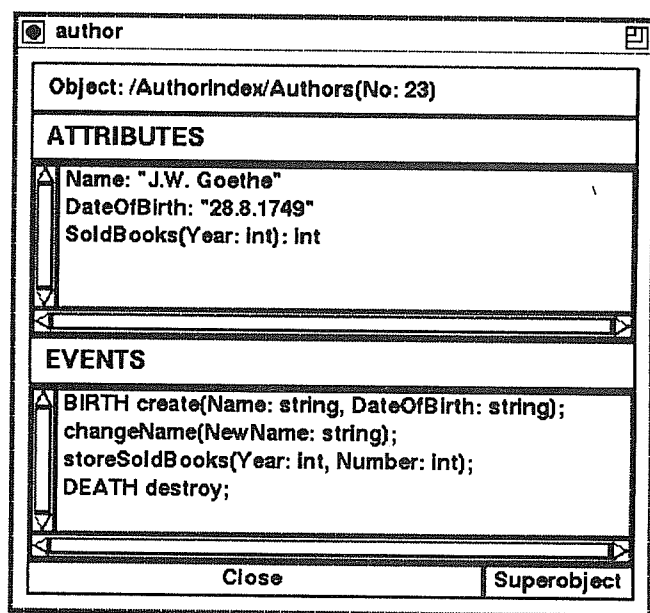
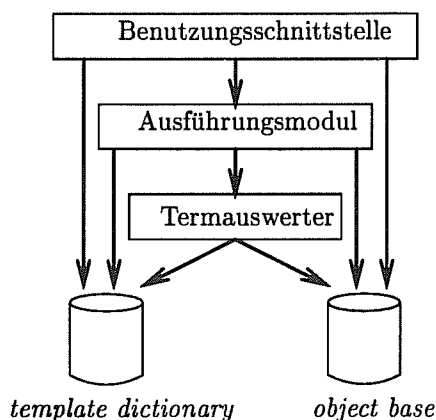


Abbildung 4: Visualisierung von Objektzuständen.

von TROLL *light*, so daß beliebige TROLL *light*-Spezifikationen in Formelmengen dieses Kalküls übersetzt werden können. Um ein möglichst einfaches Ableitungssystem zu erhalten, beschränken wir uns auf eine bestimmte Formelstruktur. Dieser Kalkül ist als Grundlage für weitere Arbeiten entworfen worden. Daher ist er sehr elementar gehalten und umfaßt in der hier vorgestellten Fassung keine über TROLL *light* hinausgehenden Konzepte.

Im folgenden werden wir den Kalkül kurz und informell skizzieren. Anschließend präsentieren wir die Verwendung des Beweisunterstützungssystems, indem wir den von uns angestrebten Arbeitsablauf beim Verifizieren von Eigenschaften darstellen.

Verifikationskalkül. Um einen Eindruck von dem Verifikationskalkül zu vermitteln, stellen wir die Hauptkonzepte anhand von Beispielen vor. Das zentrale Konzept ist das Konzept der Formeln. Wie bereits erwähnt, erlauben wir nur eine spezielle Form von Klauseln als Formeln: $P_1, \dots, P_n \rightarrow Q_1, \dots, Q_m$ wobei die P_i und Q_j sogenannte Propositionen (Aussagen) sind. Eine solche Formel ist folgendermaßen zu verstehen: In jedem Zustand (der Objektgesellschaft), in dem alle Propositionen P_1, \dots, P_n erfüllt sind, ist auch mindestens eine der Propositionen Q_1, \dots, Q_m erfüllt.



Visualisierung von Zuständen
Eingabe von Ereignismengen
Visualisierung von Zustandsübergängen

Bestimmung eines Nachfolgestands
bei gegebener Ereignismenge

Implementierung des CCV

Datenbanken zur Ablage von
Objektbeschreibungen/Objektzuständen

Abbildung 3: Architektur des TROLL *light*-Animationssystems.

Als nächstes müssen wir nun Propositionen näher betrachten. Elementare Propositionen sind Prädikatausdrücke $p(t_1, \dots, t_n)$ (mit einem Prädikatsymbol p und geeigneten Termen t_i). Darüber hinaus erlauben wir Negation von Propositionen (also $\neg P$) und die Verwendung positionaler Operatoren (d.h., $[t_o.t_e]P$, wobei t_o ein Term ist, der ein Objekt bezeichnet, und t_e ein Term, der ein Ereignis für dieses Objekt beschreibt). Eine Proposition $[t_o.t_e]P$ kann wie folgt gelesen werden: „Wenn das nächste im Objekt t_o auftretende Ereignis t_e ist, dann gilt danach P “. Wir unterscheiden zwischen zwei Arten von Prädikatsymbolen: zustandsabhängige und zustandsunabhängige. Beispielsweise sind Prädikatsymbole aus einer Datensignatur zustandsunabhängig (z.B. \leq für ganze Zahlen), während Attribute von Objekten durch zustandsabhängige Prädikate dargestellt werden. Darüber hinaus gibt es zustandsabhängige Prädikate, um die Zulässigkeit (*enable*) und das tatsächliche Auftreten (*occur*) von Ereignissen behandeln zu können.

Die folgenden Formeln beschreiben Eigenschaften von Objekten, die Personen modellieren. Als Attribute haben diese Objekte einen Namen (*Name*) und einen Familienstand (*Status*). Als Ereignis betrachten wir hier die Namensänderung (*changeName*).

$$\begin{aligned}
 & \text{Status}(P, \textit{ledig}) \rightarrow \\
 & \quad [P.\textit{changeName}(N)]\text{Status}(P, \textit{verheiratet}) \\
 & \rightarrow [A.\textit{changeName}(N)]\neg \textit{enable}(A.\textit{changeName}(N))
 \end{aligned}$$

Die erste Formel gibt an, daß das Attribut *Status*

eines Personenobjektes P nach Eintreten eines *changeName*-Ereignisses für dieses Objekt den Wert *verheiratet* annimmt, vorausgesetzt dieses Attribut hatte zuvor den Wert *ledig*. Die zweite Formel beschreibt, daß nach dem Eintreten eines *changeName*-Ereignisses in einem Personenobjekt P ein nochmaliges Eintreten eines *changeName*-Ereignisses für dieses Objekt nicht erlaubt ist.

Eine kurze Bemerkung zu der Frage, welche Arten von Eigenschaften mit diesem Verifikationskalkül bewiesen werden können, ist hier angebracht. In der hier vorgestellten Form ist die Ausdrucksfähigkeit des Kalküls (mit dem dazugehörigen Ableitungssystem) beschränkt. Wir können z.B. nicht beliebige temporale Eigenschaften für Objekte formulieren, da wir nur die positionalen Operatoren zur Verfügung haben, die in ihrer Mächtigkeit in etwa mit dem *next*-Operator temporaler Logiken vergleichbar sind. Folglich können wir z.B. auch nicht Lebendigkeitsbedingungen ausdrücken. Wir müssen uns gegenwärtig auf einzelne Zustandsübergänge oder endliche Folgen davon beschränken. Auf die Möglichkeit der späteren Erweiterungen, insbesondere um temporale Operatoren, haben wir jedoch schon bei der Basisversion dieses Kalküls geachtet.

Verifikation von Eigenschaften. Nachdem wir nun den Verifikationskalkül kurz vorgestellt haben, können wir auch einen Eindruck davon geben, wie das Beweisunterstützungssystem in die Entwicklungsumgebung eingebettet ist. Dies kann auf einfache Weise von der beabsichtigten Benutzung des

Beweisunterstützungssystem geschlossen werden. Prinzipiell stellen wir uns den folgenden Arbeitsablauf vor, den wir auch in Abb. 5 dargestellt haben:

1. Die Transformation einer TROLL *light*-Spezifikation liefert eine entsprechende Template-Signatur zusammen mit einer Menge von Formeln, die eine logische Beschreibung der spezifizierten Objekte mit Mitteln des Verifikationskalküls darstellt. Die Ergebnisse solcher Transformationen werden wiederum in einer Datenbank gespeichert.
2. Liegt das Ergebnis der Transformation einer TROLL *light*-Spezifikation vor, kann der generische Theorembeweiser ISABELLE [Pau90] aktiviert werden. Um ISABELLE als Beweiser für den Verifikationskalkül einsetzen zu können, müssen zunächst die Syntaxdefinitionen und Inferenzregeln des Verifikationskalküls und eine vorgebene Datentypsignatur mit zugehörigen Axiomen geladen werden. Als nächstes müssen die Signatur und die Formeln geladen werden, die aus der Transformation in Schritt 1. erhalten wurden.
3. Nun können wir eine zu beweisende Eigenschaft als Formel des Verifikationskalküls formulieren und den Beweiser darauf anwenden. Der Beweis kann interaktiv gesteuert werden. Wir können den Beweis Schritt für Schritt durchführen, indem wir einzelne Inferenzregeln anwenden. Wir können aber auch ISABELLE durch Wahl geeigneter Beweistaktiken veranlassen, Teile des Beweises selbständig zu führen.
4. Falls der Beweis erfolgreich durchgeführt wurde, kann die bewiesene Formel in der Datenbank abgespeichert werden. Damit kann diese Formel später wiederverwendet werden, um weitere Eigenschaften derselben Objekte zu beweisen.

Der hier vereinfacht dargestellte Ablauf kann als Grundlage für ein komfortables Beweisunterstützungssystem verwendet werden. Der realisierte Prototyp umfaßt nur die elementaren Funktionalitäten. Eine Reihe von Verbesserungen, z.B. hinsichtlich der Darstellung von Beweisen, wird not-

wendig sein, damit nicht nur wenige Spezialisten damit arbeiten können werden.

6 Schlußbemerkungen

Die Sprache TROLL *light* wurde in mehreren, verschiedenen großen Fallstudien eingesetzt [Ehr93], z.B. für die Spezifikation einer Fertigungszelle [HV94]. Hieraus läßt sich ableiten, daß unsere Sprache nicht nur für Informationssysteme geeignet ist. Die entworfene Entwicklungsumgebung ist in ihren Grundzügen prototypisch implementiert worden. Die näher vorgestellten Werkzeuge, das Animationssystem für TROLL *light*-Objektgesellschaften und das Beweisunterstützungssystem für die Verifikation von Objekteigenschaften, wurden bereits an verschiedenen kleineren Fallstudien getestet.

Auch wenn das Ziel unserer Arbeit kein marktreifes Produkt war, zeigen die erzielten Ergebnisse mit den realisierten Prototypen bereits, daß eine entsprechende Umsetzung für den realen Einsatz prinzipiell schon möglich und sinnvoll ist. Erforderlich sind dann neben einer effizienten Implementierung noch einige Verbesserungen, z.B. eine komfortablere Benutzerschnittstelle für das Beweisunterstützungssystem.

Die Ergebnisse und Erfahrungen dieses Projektes haben uns davon überzeugt, daß der objektorientierte Entwurf auch für sicherheitskritische Anwendungen geeignet ist. Die theoretische Fundierung und die Bereitstellung von Validations- und Verifikationswerkzeugen wird dabei in Zukunft eine immer größere Rolle spielen, da die immer stärker werdenden Anforderungen an Zuverlässigkeit und Korrektheit eines Informationssystems eine Verlagerung des Entwicklungsaufwandes zugunsten der frühen Entwicklungsphasen erforderlich machen.

Das objektorientierte Paradigma erlaubt eine natürliche Modellierung des zu betrachtenden Weltausschnittes und kann so dazu beitragen, komplexe Strukturen und Zusammenhänge von Objekten in Informationssystemen besser in den Griff zu bekommen. Durch Animation von spezifizierten Objektgesellschaften läßt sich unmittelbar die Adäquatheit der Spezifikation prüfen. Erforderliche formale Nachweise von Objekteigenschaften lassen sich ebenfalls frühzeitig durchführen, da — wie wir ge-

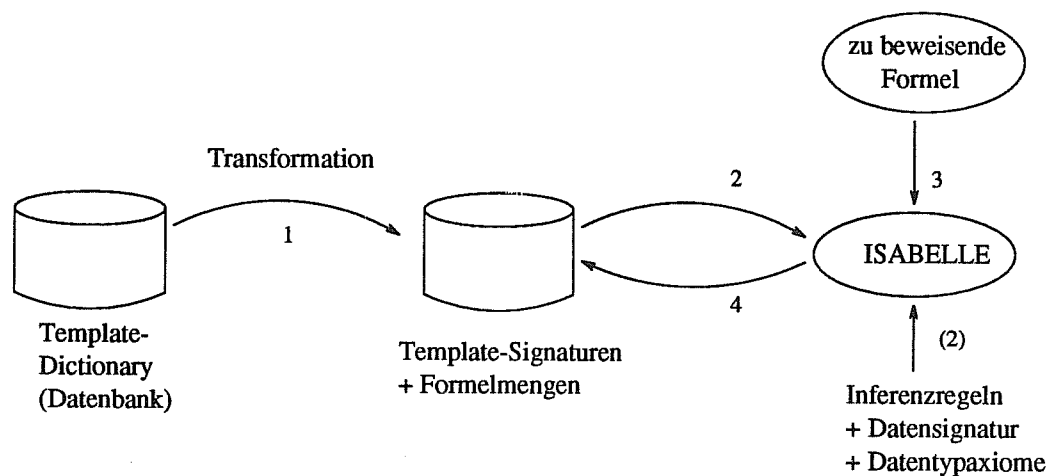


Abbildung 5: Arbeiten mit dem Beweisunterstützungssystem.

zeigt haben — geeignete Verifikationsansätze auch im objektorientierten Bereich zur Verfügung stehen. In diesem Beitrag haben wir nur einen kurzen Überblick unserer Arbeit geben können und wollen. Ausführlichere Darstellungen der hier vorgestellten Aspekte und weiterer, hier nicht berücksichtigter Aspekte finden sich in der angegebenen Literatur.

Literatur

- [BD93] A.-P. Bröhl und W. Dröschel. *Das V-Modell: Der Standard für die Softwareentwicklung mit Praxisleitfaden*. Oldenbourg-Verlag, München, 1993.
- [BJ93] M. Broy und S. Jähnichen. Das BMFT-Verbundprojekt „Korrekte Software (KOR-SO)“. *Informatik — Forschung und Entwicklung*, 8(3):152–167, 1993.
- [CDG⁺93] S. Conrad, G. Denker, M. Gogolla, R. Herzig, N. Vlachantonis und H.-D. Ehrich. Zur Entwicklung zuverlässiger Informationssysteme in KorSo. In H. Reichel, Herausgeber, *Informatik — Wirtschaft — Gesellschaft, Proc. 23. GI-Jahrestagung (GI'93)*, S. 464–469. Springer, Informatik aktuell, 1993.
- [CGH92] S. Conrad, M. Gogolla und R. Herzig. TROLL *light*: A Core Language for Specifying Objects. Informatik-Bericht 92-02, TU Braunschweig, 1992.
- [Con94] S. Conrad. On Certification of Specifications for TROLL *light* Objects. In H. Ehrig und F. Orejas, Herausgeber, *Proc. 9th Workshop on Abstract Data Types — 4th Compas Workshop (ADT'92)*, S. 158–172. Springer LNCS 785, 1994.
- [CRB93] R. Crosière, R. van de Riet und A. Blom. An Animation Facility to Simulate an Information and Communication System. In C. Roland, F. Bodart und C. Cauvet, Herausgeber *Advanced Information Systems Engineering Proc. 5th CAiSE'93*, S. 547–568. Springer Berlin, LNCS 685, 1993.
- [DHS⁺91] S. Dewal, H. Hormann, L. Schöppe, U. Kelter, D. Platz und M. Roschewski. Bewertung von Objektmanagementsystemen für Software-Entwicklungsumgebungen. In H.-J. Appelrath, Herausgeber, *Proc. Datenbanksysteme in Büro, Technik und Wissenschaft (BTW'91)*, S. 404–411. Springer, Berlin, Informatik-Fachberichte 270, 1991.
- [Ehr93] H.-D. Ehrich, Herausgeber. *Beiträge zu KORSo- und TROLL light- Fallstudien*. Technische Universität Braunschweig, Informatik-Bericht, 93-11, 1993.
- [GCH93] M. Gogolla, S. Conrad und R. Herzig. Sketching Concepts and Computational Model of TROLL *light*. In A. Miola, Herausgeber, *Proc. 3rd Int. Conf. Design and Implementation of Symbolic Computation Systems (DISCO'93)*, S. 17–32. Springer, Berlin, LNCS 722, 1993.
- [GHC⁺94] M. Gogolla, R. Herzig, S. Conrad, G. Denker und N. Vlachantonis. Integrating the ER Approach in an OO Environment. In R. Elmas-

- ri, V. Kouramajian und B. Thalheim, Herausgeber, *Proc. 12th Int. Conf. on the Entity-Relationship Approach (ER'93)*, S. 382–395. Springer, Berlin, LNCS, 1994.
- [HCG94] R. Herzig, S. Conrad und M. Gogolla. Compositional Description of Object Communities with TROLL light. In C. Chrisment, Herausgeber, *Proc. Basque Int. Workshop on Information Technology (BIWIT'94)*, S. 183–194. Cepaduès-Éditions, Toulouse, 1994.
- [HV94] R. Herzig und N. Vlachantonis. TROLL light — Specification with a Language for the Conceptual Modelling of Information Systems. In C. Lewerentz und T. Lindner, Herausgeber, *Case Study "Production Cell": A Comparative Study in Formal Specification and Verification*, S. 231–239. FZI-Publication 1/94, Forschungszentrum Informatik, Karlsruhe (Germany), 1994.
- [KEK⁺94] K. Kurbel, S. Eicker, F. Kersten, Th. Schneider und A. Teubner. I-CASE bei der Entwicklung eines großen Informationssystem: eine Information-Engineering-Fallstudie. *Wirtschaftsinformatik*, 36(2):130–144, 1994.
- [KM93] A. Kemper und G. Moerkotte. Basiskonzepte objektorientierter Datenbanksysteme. *Informatik Spektrum*, 16(2):69–80, 1993.
- [Krü93] F. Krückeberg. Zertifizierung von Software. *Wirtschaftsinformatik*, 35(2):183–186, 1993.
- [LL93] V. Lalioti und P. Loucopoulos. Visualisation for Validation. In C. Rolland, F. Bodart und C. Cauvet, Herausgeber, *Advanced Information Systems Engineering, Proc. 5th CAiSE'93*, S. 143–164. Springer, Berlin, LNCS 685, 1993.
- [LLOW91] C. Lamb, G. Landis, J. Orenstein und D. Weinreib. The ObjectStore Database System. *Communications of the ACM*, 34(10):50–63, 1991.
- [Ous93] J.K. Ousterhout. *Tcl and the Tk Toolkit*. Computer Science Division, University of California at Berkeley, 1993.
- [Pau90] L.C. Paulson. Isabelle: The Next 700 Theorem Provers. In P. Odifreddi, Herausgeber, *Logic and Computer Science*, S. 361–385. Academic Press, 1990.
- [Pis93] P. Pistor. Objektorientierung in SQL3: Stand und Entwicklungstendenzen. *Informatik Spektrum*, 16(2):89–94, 1993.
- [Rom93] H.D. Rombach. Software-Qualität und Qualitätssicherung. *Informatik Spektrum*, 16(5):267–272, 1993.
- [VH93] N. Vlachantonis und P. Hartel. An Approach towards the Conceptual Modelling of the Tool Integration Aspect. In J.-C. Rault, Herausgeber, *Proc. 6th Int. Conf. Software Engineering & Its Applications (SEA'93)*, S. 543–553. EC2, Nanterre, France, 1993.
- [VHG⁺93] N. Vlachantonis, R. Herzig, M. Gogolla, G. Denker, S. Conrad und H.-D. Ehrlich. Towards Reliable Information Systems: The CORSO Approach. In C. Rolland, F. Bodart und C. Cauvet, Herausgeber, *Proc. 5th Int. Conf. on Advanced Information Systems Engineering (CAiSE'93)*, S. 463–482. Springer, Berlin, LNCS 685, 1993.

Boolesch Algebraische Generalisierung / Spezialisierung von Objekttypen (BAGS-Modellierung)

Gisbert Englmeier
Dipl.-Betriebswirt (FH)
Löserweg 7
74722 Buchen
Telefon/Fax: 06281/2435

ysant

- 0 Einführung
- 1 Beschreibung der Vorgehensweise
 - 1.1 Attribute und ihre Interpretation als Boolesche Variable
 - 1.2 Zweckorientierte Datensicht
 - 1.3 Konjunktive Darstellung einer zweckorientierten Datensicht
 - 1.4 Disjunktive Darstellung zweckorientierter Datensichten
 - 1.5 Graphische Interpretation der Boolesch algebraischen Klammersausdrücke
 - 1.6 Absorptionssatz (Verschmelzungsfall)
 - 1.7 Sich gegenseitig ausschließende Attribute (Disjunktion)
 - 1.8 Alternative graphische Darstellung einer Konjunktion
- 2 Hauptgeneralisierung / -spezialisierung
 - 2.1 Bestimmung der Wurzelobjekttypen bei Generalisierungen / Spezialisierungen
 - 2.2 Vernetzte Generalisierungs- / Spezialisierungsstrukturen (Mehrfachvererbung)
 - 2.3 Elementare Attribute (Variable)
 - 2.4 Mehrere gültige Lösungen bei Generalisierungen / Spezialisierungen
- 3 Instanzierungstypen (in Vorbereitung)
- 4 Systematisierte und integrierte Datenmodell- und Instanzenwartung (in Vorbereitung)
- 5 Nebengeneralisierung / -spezialisierung (in Vorbereitung)

0 Einführung
Generalisierungen / Spezialisierungen gewinnen künftig immer mehr an Bedeutung. Die heute aktuelle objektorientierte Vorgehensweise in der Anwendungssystementwicklung verlangt korrekte Klassenhierarchien zur Vererbung von Attributen und Prozessen. Aber auch in herkömmlichen Methoden ist eine saubere Generalisierung / Spezialisierung Voraussetzung für qualitativ hochwertige Anwendungssysteme.

Die hier beschriebene Vorgehensweise ermöglicht es, Generalisierungen / Spezialisierungen formal und jederzeit reproduzierbar zu „berechnen“. Die Attribute von zu generalisierenden / spezialisierenden Objekttypen werden mit Operationen der Booleschen Algebra bearbeitet. Bestimmte Elemente der Booleschen Algebra werden problemadäquat interpretiert. Ausgangsbasis für die Vorgehensweise ist ein Datenmodell, das vorgegebene zweckorientierte Datensichten enthält.

Je nach Problemstruktur erhält man mehrere Lösungen, die aber alle Boolesch algebraisch korrekt sind und damit in das Datenmodell als Generalisierungen / Spezialisierungen übernommen werden können. Falls man mehrere Lösungen erhält, können verschiedene Aspekte zur Bestimmung optimaler Strukturen herangezogen werden. Während der Hauptgeneralisierung / -spezialisierung entsteht eine strenge, baumartige Hierarchie. Erfordert die Problemstellung eine Netzstruktur, wird diese in der Nebengeneralisierung / -spezialisierung ermittelt.

Ein wesentliches Merkmal der Vorgehensweise ist ein geregeltes Warnungskonzept für generalisierte- / spezialisierte Datenmodelle. Das Datenmodell „altert“ damit nicht durch laufende Änderungen oder Anpassungen. Es sind keine „Notlösungen“ in Form von „Overriden“ - Attributen notwendig. Vorliegende Instanzen [ScN690] (Ausprägungen, Exemplare) werden, wie die Modellierungskomponenten selbst, vom alten in den neuen Zustand überführt.

Boolesch Algebraische Generalisierung / Spezialisierung von Objekttypen (BAGS-Modellierung) von Gisbert Englmeier
stand überführt. Darüber hinaus könnte es notwendig sein, den alten Zustand neben dem neuen gleichzeitig im Datenbanksystem operabel zu halten und durch versionierte Prozeduren mit entsprechenden Datenreihen unterschiedlich zu bearbeiten.

Die mit dieser Methode entwickelten referentiellen Integritätsbedingungen zwischen den generalisierten / spezialisierten Objekttypen können mit Hilfe eines aktiven Data Dictionary Systems dynamisch während der Ausführungszeit überwacht werden. Ebenso ist es möglich, den Konstruktionsprozess des Datenmodells zu automatisieren. Es sind erhebliche Fortschritte in der Produktivität bei der Entwicklung von Datenmodellen und in der Erhaltung der Integrität einer Datenbank während ihres Betriebes zu erwarten.

1 Beschreibung der Vorgehensweise

In dieser Vorgehensweise ist der organisatorisch existentielle Aspekt von Attributen [ScN690] und deren Zuordnung zu zweckorientierten Datensichten von Interesse. In einer zweckorientierten Datensicht werden Attribute zusammengefasst, die durch einen detaillierten Geschäftsvorgang im Zusammenhang benötigt werden. Attributinhalt (Wertebereiche) werden nur am Rande behandelt. Nachdem die zweckorientierten Datensichten in die disjunktive Normalform [Bart77] [InDu89] überführt wurden, können sie durch Anwendung des Distributivgesetzes generalisiert / spezialisiert werden. Daraus werden die Generalisierungen / Spezialisierungen durch Anwendung des Distributivgesetzes gewonnen. Die sich ergebenden Klammersausdrücke können graphisch in einer gewünschten Notation interpretiert werden. Sie sichern die referentiellen Integritäten zwischen den einzelnen Objekttypen präzise ab.

Ziel ist es, die Ordnungsstruktur der Attribute unter Beibehaltung der Semantik der zweckorientierten Datensichten so zu gestalten, daß jedes Attribut in den generalisierten / spezialisierten Objekttypen [ScN690] nur einmal vorkommt. Vom Umfang her werden die Attribute, je nach der Abgrenzung des zu untersuchenden Systems, global (z.B. unternehmensweit) oder nur lokal (z.B. bereichsweit) bearbeitet.

Diese Beschreibung behandelt in den praktischen Beispielen zweckorientierte Datensichten nur bereichsweit. Die Thematik wird von der pragmatischen Seite her aufgearbeitet. Insoweit mathematische Konzepte der Booleschen Algebra zweckdienlich sind, werden sie problemadäquat interpretiert und in das Konzept der Problemlösung eingebaut. Die Vorgehensweise wird als „Boolesch Algebraische Generalisierung / Spezialisierung von Objekttypen“ bezeichnet („BAGS“-Modellierung).

Diese Art der Vorgehensweise ist nicht nur auf Objekttypen beschränkt. So können betriebliche Prozesse in gleicher Weise strukturiert werden, um z.B. funktionale Redundanz zu vermeiden. Ebenso können Ordnungssysteme in anderen Bereichen mit Hilfe dieser Vorgehensweise eindeutig strukturiert werden. Für alle Instanzen [ScN690] (Ausprägungen von Objekttypen, Exemplare) der generalisierten / spezialisierten Objekttypen empfiehlt es sich, Eindeutigkeit durch Benutzung entsprechender Schlüsselssysteme herzustellen.

1.1 Attribute und ihre Interpretation als Boolesche Variable

In dieser Vorgehensweise wird ein Attribut als eine Boolesche Variable aufgefaßt, die den Wert „0“ oder „1“ annehmen kann. Der Wert „1“ bedeutet, daß ein Attribut in die Generalisierung / Spezialisierung einbezogen wird. Ein nicht in die Generalisierung / Spezialisierung einbezogenes Attribut ist mit dem Wert „0“ belegt. Es existiert damit nicht in zu untersuchenden System und wird deshalb weder erwähnt noch in die Bearbeitung einbezogen. Alle zu einem zu untersuchenden System gehörenden Attribute existieren somit explizit berechtigt. Sie sind mit dem Wert „1“ (existent) belegt.

Ein Attribut ist systemweit eindeutig festzulegen. Die Anzahl n der Attribute eines Systems ist endlich. Damit eine kurze aber dennoch aussagefähige Notation der Attribute vorgenommen werden kann, wird in dieser Abhandlung ein Attribut (Boolesche Variable) neben seinem jeweiligen Attributnamen auch als Variable a_i (mit $i = 1, 2, \dots, n$) benannt.

Folgende Tabelle zeigt ein abstraktes Beispiel mit Attributen:

| LfdNr | Attribute (Boolesche Variable) | Attribut (abgekürzt) | Wertzuweisung |
|-------|--------------------------------|----------------------|---------------|
| | 1 | 2 | j |
| 1 | Attribut a | a ₁ | =1 |
| 2 | Attribut b | a ₂ | =1 |
| 3 | Attribut c | a ₃ | =1 |
| 4 | Attribut d | a ₄ | =1 |
| 5 | Attribut e | a ₅ | =1 |

Tabelle 1: Systemweite Attributtabelle (Beispiel mit abstrakten Attributen)

Alle Booleschen Variablen (Attribute), die ein zu untersuchendes System repräsentieren, bilden eine Menge M. Sie sind untereinander konjunktiv (mit „Logisch - Und“, Logische Multiplikation) zu verknüpfen. Diesem Boolesch algebraischen Produkt kann der Funktionswert „1“ zugewiesen werden, falls alle Booleschen Variablen den Wert „1“ besitzen, d.h. daß sie existieren. Dies bedeutet auch, daß für jedes Attribut ein gültiger Wertebereich existiert. Jeder Wert des Wertebereichs muß genau der Semantik des Attributes genügen. So dürfen z.B. beim Attribut „Steuerklasse eines Arbeitnehmers“ nur die tatsächlichen Steuerklassen 1 bis 6 vorkommen. Die Werte „leer“, „blank“, „keine Steuerklasse“ oder „0“ dürfen z.B. nicht im Wertebereich existieren.

Es gilt somit für die Attribute aus Tabelle 1:

$$\text{Attribut a} \wedge \text{Attribut b} \wedge \text{Attribut c} \wedge \text{Attribut d} \wedge \text{Attribut e} = 1$$

$$a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge a_5 = 1$$

oder als „Logische Multiplikation“

$$a_1 * a_2 * a_3 * a_4 * a_5 = 1$$

oder kürzer als „Logische Multiplikation“

$$a_1 a_2 a_3 a_4 a_5 = 1$$

$$\text{mit } a_1 = a_2 = a_3 = a_4 = a_5 = 1.$$

Bemerkung: Für die folgenden „Logisch - Und“ Ausdrücke wird, von Ausnahmen abgesehen, die Form, wie sie in (1.-4) gezeigt wird, verwendet.

Konkretes Beispiel mit Attributen:

| Lfd. Nr. | Attribute (Boolesche Variable) | Attribut (abgekürzt) | Wertzuweisung |
|----------|--------------------------------|----------------------|---------------|
| 1 | KundenNr | a ₁ | =1 |
| 2 | Name | a ₂ | =1 |
| 3 | Anschrift | a ₃ | =1 |
| 4 | Bonität | a ₄ | =1 |
| 5 | Kreditgrenze | a ₅ | =1 |
| 6 | LieferantNr | a ₆ | =1 |
| 7 | PersonalNr | a ₇ | =1 |
| 8 | Steuerklasse | a ₈ | =1 |
| 9 | Finanzamt | a ₉ | =1 |
| 10 | Abteilung | a ₁₀ | =1 |
| 11 | Stundensatz | a ₁₁ | =1 |
| 12 | Lohngruppe | a ₁₂ | =1 |

Tabelle 2: Systemweite Attributtabelle (Beispiel mit konkreten Attributen)

Die Attribute aus Tabelle 2 sind untereinander „Logisch Und“ zu verknüpfen:

$$\text{KundenNr} \wedge \text{Name} \wedge \text{Anschrift} \wedge \text{Bonität} \wedge \text{Kreditgrenze} \wedge \text{LieferantNr} \wedge \text{PersonalNr} \wedge \text{Steuerklasse} \wedge \text{Finanzamt} \wedge \text{Abteilung} \wedge \text{Stundensatz} \wedge \text{Lohngruppe} = 1 \quad (2.1)$$

oder verkürzt

$$a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 a_{10} a_{11} a_{12} = 1 \quad (2.2)$$

oder

$$\bigwedge_{i=1}^{12} a_i = \prod_{i=1}^{12} a_i = 1 \quad (2.3)$$

Für alle n Attribute a_i eines zu untersuchenden Systems gilt:

$$\bigwedge_{i=1}^n a_i = \prod_{i=1}^n a_i = 1 \quad (2.4)$$

Alle im System benutzten Attribute müssen ihrer Semantik entsprechend vorkommen und vorgesehen sein. Dies sagt noch nichts über tatsächlich vorkommende Instanzen mit den benutzten Attributen aus.

1.2. Zweckorientierte Datensicht

Für verschiedene betriebliche Aspekte oder Vorgänge werden Attribute des zu untersuchenden Systems zu zweckorientierten Datensichten zusammengefaßt und in Zuordnungsmatrizen als Attributteilungen der Menge M ausgewiesen. Ein „+“-Zeichen in einem Matrixelement einer Zuordnungsmatrix von Attributen zu zweckorientierten Datensichten bedeutet, daß das entsprechende Attribut zur jeweiligen zweckorientierten Datensicht eines Bearbeitungsvorgangs gehört bzw. einem betrieblichen Aspekt entsprechend zugeordnet ist. Ist ein Matrixelement leer, gehört das in dieser Zeile referenzierte Attribut nicht zu der in dieser Spalte referenzierten zweckorientierten Datensicht.

Abstraktes Beispiel:

| LfdNr | Attribute (Boolesche Variable) | Attribute (abgekürzt) | Zweckorientierte Datensicht S ₁ | Zweckorientierte Datensicht S ₂ |
|-------|--------------------------------|-----------------------|--|--|
| | 1 | 2 | 3 | 4 |
| 1 | Attribut a | a ₁ | + | + |
| 2 | Attribut b | a ₂ | + | + |
| 3 | Attribut c | a ₃ | + | + |
| 4 | Attribut d | a ₄ | + | + |
| 5 | Attribut e | a ₅ | | + |

Tabelle 3: Tabelle der zweckorientierten Datensichten A und B mit Attributzuweisung

Ein konkretes Beispiel mit den zweckorientierten Datensichten Kunde, Lieferant und Gewerblicher Arbeitnehmer und den ausgewählten Attributen Kundennummer, Name, Anschrift, Bonität, Kreditgrenze, Lieferantnummer, Personalnummer, Steuerklasse, Finanzamt, Abteilung, Stundensatz für gewerbliche Arbeitnehmer und Lohngruppe für gewerbliche Arbeitnehmer zeigt die folgende Tabelle.

Die Zuordnung der Attribute ist in den Spalten drei bis fünf durch den Eintrag eines „+“- Zeichens gekennzeichnet.

| Lfd. Nr. | Attribute (Boolesche Variable) | Attribute (abgekürzt) | Zweckorientierte Datensichten | | |
|----------|--------------------------------|-----------------------|-------------------------------|----------------|-----------------------|
| | | | Kunde | Lieferant | Gewerbl. Arbeitnehmer |
| | 1 | 2 | S ₁ | S ₂ | S ₃ |
| 1 | KundenNr | a ₁ | 3 | 4 | 5 |
| 2 | Name | a ₂ | + | + | + |
| 3 | Anschrift | a ₃ | + | + | + |
| 4 | Bonität | a ₄ | + | | |
| 5 | Kreditgrenze | a ₅ | + | | |
| 6 | LieferantNr | a ₆ | | + | + |
| 7 | PersonalNr | a ₇ | | | + |
| 8 | Steuerklasse | a ₈ | | | + |
| 9 | Finanzamt | a ₉ | | | + |
| 10 | Abteilung | a ₁₀ | | | + |
| 11 | Stundensatz | a ₁₁ | | | + |
| 12 | Lohngruppe | a ₁₂ | | | + |

Tabelle 4: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“ und „Gewerblicher Arbeitnehmer“

1.3 Konjunktive Darstellung einer zweckorientierten Datensicht

Alle Attribute (Booleschen Variablen), die eine zweckorientierte Datensicht repräsentieren, sind untereinander konjunktiv (mit „Logisch-Und“) zu verknüpfen. Jedem Boolesch algebraischen Produkt kann der Funktionswert „1“ zugewiesen werden, falls alle zu einer zweckorientierten Datensicht gehörenden Booleschen Variablen existieren (den Wert „1“ besitzen).

Abstraktes Beispiel Tabelle 3:

$$S_1 = a_1 \cdot a_2 \cdot a_3 \cdot a_4 = 1 \quad (3.1)$$

$$S_2 = a_1 \cdot a_2 \cdot a_5 = 1 \quad (3.2)$$

mit $a_1 = a_2 = a_3 = a_4 = a_5 = 1$.

Konkretes Beispiel aus Tabelle 4:

$$S_1 = \text{Kunde} = \text{KundenNr} \wedge \text{Name} \wedge \text{Anschrift} \wedge \text{Bonität} \wedge \text{Kreditgrenze} = 1 \quad (4.1)$$

$$S_2 = \text{Lieferant} = \text{Name} \wedge \text{Anschrift} \wedge \text{LieferantNr} = 1 \quad (4.2)$$

$$S_3 = \text{Gewerblicher Arbeitnehmer} = \text{Name} \wedge \text{Anschrift} \wedge \text{PersonalNr} \wedge \text{Steuerklasse} \wedge \text{Finanzamt} \wedge \text{Abteilung} \wedge \text{Stundensatz} \wedge \text{Lohngruppe} = 1 \quad (4.3)$$

oder verkürzt:

$$S_1 = \text{Kunde} = a_1 \cdot a_2 \cdot a_3 \cdot a_4 \cdot a_5 = 1 \quad (4.4)$$

$$S_2 = \text{Lieferant} = a_2 \cdot a_3 \cdot a_6 = 1 \quad (4.5)$$

$$S_3 = \text{Gewerblicher Arbeitnehmer} = a_2 \cdot a_3 \cdot a_7 \cdot a_8 \cdot a_9 \cdot a_{10} \cdot a_{11} \cdot a_{12} = 1 \quad (4.6)$$

$$\text{mit } a_1 = a_2 = a_3 = a_4 = a_5 = a_6 = a_7 = a_8 = a_9 = a_{10} = a_{11} = a_{12} = 1$$

Durch die Erfüllung dieser Gleichungen kommt inhaltlich zum Ausdruck, daß die jeweilige zweckorientierte Datensicht nur dann berechnigt existiert, wenn in ihren vorkommenden Instanzen (Ausprägungen) alle Attribute in jedem Fall immer mit Werten aus ihren jeweiligen Wertebereichen belegt sind. Dabei ist zu beachten, daß die Werte der bedachten Semantik des Attributes entsprechen müssen. Es entfallen somit Fallbehandlungen in

Verarbeitungsprozessen, um die tatsächliche Datensicht während des Prozessablaufs dynamisch zu erzeugen. Aus den dynamischen Verarbeitungsprozessen wird die intensive Datensichtselektionsverarbeitung herausgenommen und in das statische Datenmodell verlagert. Der bei solchen intensiven Prozessen notwendige Testaufwand wird damit erheblich reduziert.

Eine zweckorientierte Datensicht ist entweder vollständig (mit allen Attributen belegt) zu verwenden oder sie existiert nicht.

Für eine zweckorientierte Datensicht S_k gilt:

$$S_k = \bigwedge_{i \in J_k} a_i = 1 \quad (4.7)$$

k: Laufindex der zweckorientierten Datensichten

J_k: Menge der in einer zweckorientierten Datensicht S_k benötigten Indizes i der Attribute a

i: Systemweit eindeutiger Index für jedes Attribut a

a_i: Attribut a, eindeutig identifiziert durch Index i

Die maximale Anzahl p der zweckorientierten Datensichten S_k ergibt sich aus der Potenzmenge der Menge M aller Attribute ohne die leere Menge und ohne die Menge M selbst. Die Attribute a_i jeder zweckorientierten Datensicht bilden eine echte Teilmenge der Menge M aller Attribute.

Entspricht ein Attribut in einer Instanz einmal nicht der bedachten Semantik, dann ist die entsprechende Boolesche Variable 0. Damit wird der gesamte Ausdruck 0, d.h. für diese Instanz gibt es eigentlich keine vorgesehene Datensicht. Durch diese strenge Vorschrift werden die in verschiedenen Methoden / Vorgehensweisen vorkommenden sogenannten optionalen Attribute [Hohe93] nicht zugelassen. Falls solche optionalen Attribute in einer Datensicht vorkommen, handelt es sich im Sinne dieser Vorgehensweise um eine nicht korrekte (unvollständige) Abbildung der Realität.

Folgendes Beispiel soll diesen Sachverhalt verdeutlichen: In einer konkreten Instanz (Ausprägung) fehlen bei der Datensicht „Gewerblicher Arbeitnehmer“ die beiden Werte für Steuerklasse und Finanzamt, weil sie beim vorliegenden Mitarbeiter nicht relevant sind. Alle anderen Merkmale stimmen mit der Datensicht „Gewerblicher Arbeitnehmer“ überein. Damit wären für diesen Fall die beiden Booleschen Variablen (aus Tabelle 4) a₈ und a₉ = 0. Für diese Instanz gibt es somit keine definierte zweckorientierte Datensicht, da die obige Ausdruck (4.3) bzw. (4.6) zu Null wird.

| Lfd. Nr. | Attribute (Boolesche Variable) | Attribute (verkürzt) | Zweckorientierte Datensichten | | |
|----------|--------------------------------|----------------------|-------------------------------|----------------|-----------------------|
| | | | Kunde | Lieferant | Gewerbl. Arbeitnehmer |
| | 1 | 2 | S ₁ | S ₂ | S ₃ |
| 1 | KundenNr | a ₁ | 3 | 4 | 5 |
| 2 | Name | a ₂ | + | + | + |
| 3 | Anschrift | a ₃ | + | + | + |
| 4 | Bonität | a ₄ | + | | |
| 5 | Kreditgrenze | a ₅ | + | | |
| 6 | LieferantNr | a ₆ | | + | + |
| 7 | PersonalNr | a ₇ | | | + |
| 8 | Steuerklasse | a ₈ | | | + |
| 9 | Finanzamt | a ₉ | | | + |
| 10 | Abteilung | a ₁₀ | | | + |
| 11 | Stundensatz | a ₁₁ | | | + |
| 12 | Lohngruppe | a ₁₂ | | | + |

Tabelle 5: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerblicher Arbeitnehmer“ und „Geringfügig Beschäftigter“

Das Fehlen der beiden Werte kann bei der Anwendung von Operationen (objektorientiert: Methoden, Services) auf diese Attribute zu unvorhergesehenen Fehlern führen. Ursache für diesen Zustand ist ein Konstruktionsfehler der zweckorientierten Datensicht „Gewerblicher Arbeitnehmer“. Sie kann keine Instanzen von Mitarbeitern repräsentieren, die fachlich gesehen kein persönlich zu versteuerndes Einkommen beziehen. Es handelt sich dabei um Mitarbeiter, die mit ihrem Einkommen unter der Mindesteinkommensgrenze liegen. Für sie ist vom beschäftigten Unternehmen eine pauschale Lohnsteuer an das Finanzamt zu entrichten ist. Deshalb ist eine weitere zweckorientierte Datensicht „Geringfügig Beschäftigter“ einzuführen, die genau die Attribute enthält, die bei dieser Datensicht relevant sind. Die Tabelle 5 ist um diese Datensicht erweitert. Damit ist es möglich, „Geringfügig Beschäftigte“ semantisch korrekt im Modell abzubilden.

1.4 Disjunktive Darstellung zweckorientierter Datensichten

Die als Konjunktionen vorliegenden zweckorientierten Datensichten des zu bearbeitenden Systems sind untereinander disjunkt („inklusive Logisch Oder“, Logische Addition) zu verknüpfen. Sie können jetzt einer Generalisierung / Spezialisierung unterzogen werden, indem die gemeinsamen als Boolesche Variable vorliegenden Attribute ausgeklammert (herausfaktoriert [Mart88]) werden. Durch Zuweisung des Funktionswertes „1“ zur entstandenen Disjunktion erhält man die disjunktive Normalform des Gesamtsystems. Der entstandene Ausdruck ist anschließend mit Hilfe des Distributivgesetzes der Booleschen Algebra zu bearbeiten. Durch das Ausklammern erhält man originalmäßig die Generalisierungs- / Spezialisierungsstrukturen:

Abstraktes Beispiel aus (3.1) und (3.2):

$$S_1 \vee S_2 = \tag{5.1}$$

$$a_1 \wedge a_2 \wedge a_4 \vee a_1 \wedge a_2 \wedge a_5 = 1 \tag{5.2}$$

Durch Anwendung des Distributivgesetzes auf den obigen Ausdruck (Ausklammern der gemeinsamen Faktoren $a_1 a_2$) erhält man:

$$a_1 a_2 (a_4 \vee a_5) = 1 \tag{5.3}$$

Damit ist die Generalisierung / Spezialisierung vollzogen. Die beiden Booleschen Variablen außerhalb der Klammer gelten bezüglich der drei Booleschen Variablen innerhalb der Klammer als generalisiert. Die drei befindlichen Booleschen Variablen als spezialisiert.

Es findet eine neue Zuweisung der Attribute bzw. der sie repräsentierenden Booleschen Variablen zu den jetzt generalisierten / spezialisierten Objekttypen statt. Aus den zwei zweckorientierten, nicht generalisierten / spezialisierten Datensichten S_1 und S_2 sind die drei generalisierten / spezialisierten Objekttypen A, B und C mit den folgenden Attributzusweisungen entstanden:

$$A = a_1 a_2, B = a_3 a_4 \text{ und } C = a_5 \tag{5.4}$$

Damit gilt gemäß Ausdruck (5.3):

$$A (B \vee C) = 1 \tag{5.5}$$

Konkretes Beispiel nach (4.1) bis (4.6):

$$S_1 \vee S_2 \vee S_3 = \tag{6.0}$$

$$\text{Kunde} \vee \text{Lieferant} \vee \text{Gewerblicher_Arbeitnehmer} = \tag{6.1}$$

$$a_1 \wedge a_2 \wedge a_3 \wedge a_5 \vee a_2 \wedge a_3 \wedge a_4 \vee a_2 \wedge a_3 \wedge a_4 \wedge a_5 \wedge a_1 \wedge a_2 = 1 \tag{6.2}$$

Durch Ausklammern von $a_2 a_3$ erhält man folgenden Ausdruck:

$$a_2 a_3 (a_1 a_5 \vee a_4 \vee a_1 a_2 a_3 a_4 \wedge a_1 a_2) = 1 \tag{6.3}$$

Eine Zuweisung der Attribute zu den entstandenen generalisierten / spezialisierten Objekttypen ergibt:

$$\text{Externer Partner} = E = a_2 a_3 \tag{6.4}$$

$$\text{Kunde} = K = a_1 a_2 a_5 \tag{6.5}$$

$$\text{Lieferant} = L = a_4 \tag{6.6}$$

$$\text{Gewerbl.Arbeitnehmer} = G = a_1 a_2 a_3 a_4 \wedge a_1 a_2 \tag{6.7}$$

Der Boolesche Ausdruck (6.3) kann damit wie folgt geschrieben werden:

$$E (K \vee L \vee G) = 1 \tag{6.8}$$

Die generalisierten / spezialisierten Objekttypen enthalten jetzt nur noch ihre spezifischen Attribute. Für zusätzlich entstehende Teilterme (Konjunktionen) sind treffende Bezeichnungen zu einzuführen. So ist für die beiden Attribute a_2 und a_3 , die durch den Generalisierungsvorgang als eigener Objekttyp isoliert wurden, eine Objekttypbezeichnung (z.B. „Externer Partner“) einzuführen. Bei Bedarf können zusätzlich neue, identifizierende Attribute eingeführt werden, z.B. die „Externe Partner“ Nummer a_0 für den Objekttyp E. Dieses Attribut kann einfach den bereits in diesem Objekttyp befindlichen Attributen konjunktiv hinzugefügt werden: $E = a_0 a_2 a_3$. Da es sich beim genannten Beispiel multiplikativ mit allen in den Klammern befindlichen Attributen verknüpft, wird es jedem spezialisierten Objekttyp vererbt.

Durch die Spezialisierungen von Attributen wird ein Symbol (ein Name) gebildet. Jeder Teilmenge (spezialisierter Objekttyp) wird ein Symbol (ein Name) zugeteilt, unter dem sie künftig explizit genannt werden kann. Damit entfällt die ansonsten notwendige dynamische Bildung dieser Teilmengen (Schritt- und/oder Vereinigungsmengen) als dynamische Datensichten während des Bearbeitungsprozesses auf Instanzebene. Die während des Bearbeitungsprozesses dynamisch gebildeten Teilmengen sind anonym (ohne Namen) und nicht dauerhaft (nicht persistent). Der Testaufwand ist erheblich. Durch die Generalisierung / Spezialisierung wird die Komplexität der Bearbeitungsprozesse wesentlich reduziert. Der formale Gehalt wird in das statische Datenmodell verlagert.

Für alle zweckorientierten Datensichten S des untersuchten Systems gilt:

$$S = \bigvee_{k=1}^p \bigvee_{i \in J_k} S_k = \bigvee_{k=1}^p \bigwedge_{i \in J_k} a_i = 1 \tag{6.9}$$

wobei p die Anzahl aller zweckorientierten Datensichten im untersuchten System ist.

Als Beispiel der Entwicklung von (6.9) sei der Ausdruck (5.2) mit $J_1 = \{1,2,3,4\}$ und $J_2 = \{1,2,5\}$ und $p = 2$ angeführt:

$$S = \bigvee_{k=1}^2 \bigvee_{i \in J_k} S_k = (\bigwedge_{i \in J_1} a_i) \vee (\bigwedge_{i \in J_2} a_i) = (a_1 \wedge a_2 \wedge a_3 \wedge a_4) \vee (a_1 \wedge a_2 \wedge a_5) = 1 \tag{6.10}$$

1.5 Graphische Interpretation der Booleschen algebraischen Klammernausrücke

Die Umsetzung eines Klammernausrucks in eine graphische Interpretation ist wie folgt. Jede „Klammer auf“ begründet eine Spezialisierung. Die dazu paarige Klammer in der gleichen Ebene beendet die Spezialisierung in dieser Ebene. Alle Booleschen Variablen (stellvertretend für die Attribute) innerhalb einer Klammer gelten bezüglich der Booleschen Variablen außerhalb der Klammer als spezialisiert. Die Variablen außerhalb einer Klammer gelten bezüglich der Booleschen Variablen innerhalb der Klammer als generalisiert.

Die Vorgehensweise ist nicht an eine der bekannten Datenmodellierungsmethoden (z.B. ERM [Sche90], SERM [FeS93], OOA nach Coad und Yourdon [CoYo90]) gebunden. Die Ergebnisse können in jeder Methode interpretiert werden. Durch die präzise Ausformulierung wird dem „semantischen Bedarf“ an Generalisierung / Spezialisierungen der unterschiedlichen Methoden Rechnung getragen. Inwieweit die jeweiligen syntaktischen Ausdrucksformen der Methoden die von dieser Vorgehensweise gelieferten Inhalte zu transportieren vermögen, ist von Methode zu Methode verschieden.

Die Darstellungen in dieser Beschreibung erfolgen stellvertretend immer in der Notation der OOA nach Coad und Yourdon, die um die ermittelten Boolesch algebraischen Ausdrücke zur Kontrolle der referenziellen Integritäten zwischen den generalisierten / spezialisierten Objekttypen erweitert wurde.

Für den Klammersausdruck des abstrakten Beispiels (5.5) als Objekttypen- bzw. (5.3) Attributausdruck mit den Attributen aus Tabelle 3 ergibt sich folgende Darstellung:

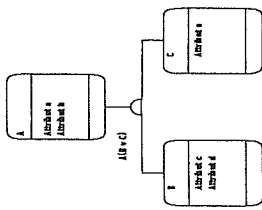


Abbildung 1: Graphische Umsetzung des Ausdrucks A(B ∨ C)

Der Klammersausdruck des konkreten Beispiels (6.8) als Objekttypen- bzw. (6.3) als Attributausdruck mit den Attributen aus Tabelle 4 führt zu folgender Darstellung:

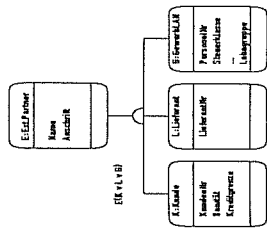


Abbildung 2: Graphische Umsetzung des konkreten Beispiels (6.8)

Die Elemente der Spezialisierung (untergeordnete Ebene) sind mit dem Element der Generalisierung (übergeordnete Ebene) konjunktiv (multiplikativ) verknüpft (vertikale Verknüpfung), z.B. E(K ∨ L ∨ G). Die Elemente der Spezialisierung (der untergeordneten Ebene) sind untereinander disjunktiv verknüpft (horizontale Verknüpfung), z.B. E(K ∨ L ∨ G). Nach einer „Klammer auf“ oder vor einer „Klammer zu“ in der gleichen Klammer Ebene muß mindestens eine Boolesche Variable stehen, da jede „Klammer auf“ eine Spezialisierung bedeutet. Überschüssige bzw. mehrfache Klammern, wie z.B. a₁((a₂(a₃))), stören bei der Umsetzung eines Ausdrucks in das graphische Modell und sind deshalb wegzulassen.

1.6 Absorptionsfall (Verschmelzungsfall)

An folgenden Beispiel wird die Absorption [Bart77] und ihre Interpretation in dieser Vorgehensweise behandelt. Folgende zweckorientierte Datensichten S₁ und S₂ mit ihren Attributen mögen vorliegen:

$$S_1 = a_1, S_2 = a_1 a_2 \quad (7.1)$$

$$S_1 \vee S_2 = \quad (7.2)$$

$$a_1 \vee a_1 a_2 = 1 \quad (7.3)$$

Die Anwendung des Distributivgesetzes ergibt folgenden Ausdruck:

$$a_1 (1 \vee a_2) = 1 \quad (7.4)$$

Nach den Regeln der Booleschen Algebra wäre der Ausdruck in der Klammer immer erfüllt (immer = 1). Er könnte deshalb weggelassen werden, er wird absorbiert. Das Ergebnis wäre

$$a_1 = 1 \quad (7.5)$$

Bei dieser Vorgehensweise darf der Klammersausdruck nicht entfernt werden. Die „1“ in der Klammer erfüllt hier die Funktion eines Platzhalters. Dies bedeutet, a₁ kann auch allein ohne a₂ in einem Objekttyp auftreten, was leicht dem Ausdruck (7.3) entnommen werden kann.

Es ergeben sich die beiden generalisierten / spezialisierten Objekttypen A mit a₁ und B mit a₂, wobei sich die Objekttypschreibweise wie folgt ergibt:

$$A (1 \vee B) = 1.$$

Die graphische Darstellung ist wie folgt:

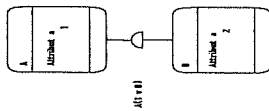


Abbildung 3: Graphische Umsetzung der Absorption

Folgendes konkrete Beispiel zeigt ebenfalls diesen Sachverhalt. Die Attribute eines Barzählers sind eine Untermenge der Attribute eines Kunden. Der Barzahler benötigt weder das Attribut Bonität noch das Attribut Kreditgrenze. Ansonsten besitzt er dieselben Attribute wie der Kunde:

| Lfd. Nr. | Attribute (Boolesche Variable) | Attribute (verkürzt) | Zweckorientierte Datensichten | | | |
|----------|--------------------------------|----------------------|-------------------------------|----------------|---------------------------|----------------|
| | | | Kunde | Lieferant | Gewerblicher Arbeitnehmer | Barzahler |
| | 1 | 2 | S ₁ | S ₂ | S ₃ | S ₄ |
| 1 | Ext.PartnerNr | a ₆ | + | + | + | + |
| 2 | KundenNr | a ₁ | + | + | + | + |
| 3 | Name | a ₂ | + | + | + | + |
| 4 | Anschrift | a ₃ | + | + | + | + |
| 5 | Bonität | a ₄ | + | | | |
| 6 | Kreditgrenze | a ₅ | + | | | |
| 7 | LieferantNr | a ₇ | | + | | |
| 8 | PersonalNr | a ₈ | | | + | |
| 9 | Steuerklasse | a ₉ | | | | + |
| 10 | Finanzamt | a ₁₀ | | | | + |
| 11 | Abteilung | a ₁₁ | | | | + |
| 12 | Stundensatz | a ₁₂ | | | | + |
| 13 | Lohngruppe | a ₁₃ | | | | + |

Tabelle 6: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerblicher Arbeitnehmer“ und „Barzahler“

Die Ausformulierung dieser Tabelle in die disjunktive Normalform ergibt folgenden Ausdruck:

$S_1 \vee S_2 \vee S_3 \vee S_4 =$ (8.0)

Kunde \vee Lieferant \vee Gewerblicher Arbeitnehmer \vee Barzahler = (8.1)

In der verkürzten Form ergibt dies:

$a_0 a_1 a_2 a_3 a_4 a_5 \vee a_0 a_2 a_3 a_4 \vee a_0 a_2 a_3 a_7 a_8 a_9 a_{10} a_{11} a_{12} \vee a_0 a_1 a_2 a_3 = 1$ (8.2)

Durch Ausklammern von $a_0 a_2 a_3 a_4$ aus Ausdruck (8.2) ergibt sich:

$a_0 a_2 a_3 (a_1 a_4 a_5 \vee a_6 \vee a_7 a_8 a_9 a_{10} a_{11} a_{12} \vee a_1) = 1$ (8.3)

Durch Ausklammern von a_1 aus Ausdruck (8.3) ergibt sich:

$a_0 a_2 a_3 (a_1 (1 \vee a_4 a_5) \vee a_6 \vee a_7 a_8 a_9 a_{10} a_{11} a_{12}) = 1$ (8.4)

Die Attribute können entsprechenden Objekttypen zugeordnet werden:

- Externer Partner = $E = a_0 a_2 a_3$ (8.5)
- Kreditur = $K = a_1$ (8.6)
- Kreditor = $R = a_4 a_5$ (8.7)
- Lieferant = $L = a_6$ (8.8)
- Gewerblicher Arbeitnehmer = $G = a_7 a_8 a_9 a_{10} a_{11} a_{12}$ (8.9)

Dies ergibt für Ausdruck (8.4) folgende Objekttypenzuordnung:

$E (K (1 \vee R) \vee L \vee G) = 1$ (8.10)

Die graphische Darstellung ergibt folgendes Bild:

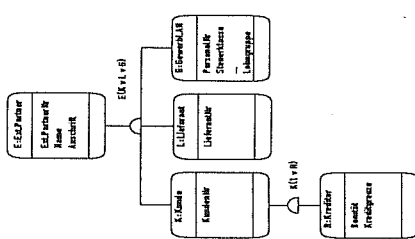


Abbildung 4: Graphische Umsetzung des mehrstufigen Ausdrucks (8.10) mit Absorption

Damit kann ein Barzahler mit seiner Untermenge an Attributen durch das System repräsentiert werden.

1.7 Sich gegenseitig ausschließende Attribute (Disjunktion)

Die explizite Abwesenheit eines Attributes in einer zweckorientierten Datensicht wird formal dadurch zum Ausdruck gebracht, daß die dieses Attribut verkörpernde Boolesche Variable negiert und der Negation der

„1“ zugewiesen wird. Die negierte Variable wird hier mit einer Unterstreichung dargestellt: \bar{a}_i . Der einer disjunktiven Normalform zugewiesene Funktionswert von „1“ bleibt erhalten. Die folgende Wahrheitstafel zeigt die Interpretation:

| Boolesche Variable | \bar{a}_1 | \bar{a}_2 |
|--|-------------|-------------|
| Wertzuweisung „Attribut existiert“ | 1 | 0 |
| Wertzuweisung „Attribut existiert nicht“ | 0 | 1 |

Tabelle 7: Wahrheitstafel bei Booleschen - Variablen „Logisch Nicht“

In einer Zuordnungsmatrix von Attributen zu zweckorientierten Datensichten wird die explizite Abwesenheit eines Attributes durch ein „-“ - Zeichen in der jeweiligen Zeile / Spalte signalisiert. Die „Logisch-Nicht“- Variablen gehen nicht mit in das Datenmodell ein. Zwei Attribute können in zwei unterschiedlichen zweckorientierten Datensichten wie folgt existieren und sich wechselseitig beeinflussen:

| LfdNr | Zweckorientierte Datensichten | | Bemerkung |
|-------|-------------------------------|-------|---|
| | S_1 | S_2 | |
| 1 | \bar{a}_1 | + | Attribut kommt in S_1 und in S_2 gleichzeitig vor |
| 2 | \bar{a}_2 | + | Attribut kommt nur in S_1 vor, unabhängig von S_2 |
| 3 | \bar{a}_3 | + | Attribut kommt nur in S_2 vor, unabhängig von S_1 |
| 4 | \bar{a}_4 | - | Attribut kommt in S_1 vor, nicht aber in S_2 |
| 5 | \bar{a}_5 | - | Attribut kommt in S_2 vor, nicht aber in S_1 |

Tabelle 8: Tabelle sich wechselseitig beeinflussender Attribute in zweckorientierten Datensichten

Zu lfd. Nr. 1 in Tabelle 8:

Das Attribut a_1 kommt immer in beiden zweckorientierten Datensichten S_1 und S_2 vor. Das Attribut a_2 wird in diesem Fall bezüglich der beiden zweckorientierten Datensichten immer vor die Klammer gezogen d.h. generalisiert werden können.

Zu lfd. Nr. 2 und 3 in Tabelle 8:

Es kann eine Instanz (Ausprägung) der zweckorientierten Datensicht S_1 mit dem Attribut a_2 existieren und/oder es kann eine Instanz (Ausprägung) der zweckorientierten Datensicht S_2 mit dem Attribut a_3 existieren. Befinden sich beide Attribute über ein drittes generalisiertes Attribut (z.B. a_1) gemeinsam in einer Klammer, gelten sie als zu diesem Attribut spezialisiert: $a_1 a_2 \vee a_1 a_3$ mit $a_1 =$ Geschäftspartner, $a_2 =$ Kunde und $a_3 =$ Lieferant. Ein Geschäftspartner kann entweder Kunde oder Lieferant oder beides gleichzeitig sein.

Zu lfd. Nr. 4 und 5 in Tabelle 8:

Falls die zweckorientierte Datensicht S_1 mit dem Attribut a_4 instanziiert (ausgeprägt) ist, darf nicht gleichzeitig die zweckorientierte Datensicht S_2 mit dem Attribut a_5 instanziiert (ausgeprägt) sein und umgekehrt. Diese gegenseitigen Ausschlüsse treten paarweise auf. Dazu müssen beide zweckorientierten Datensichten gemeinsame Attribute (z.B. a_1) besitzen, die vor die Klammer gezogen werden können. Dann ergibt sich nach der Durchführung der Generalisierung / Spezialisierung ein exklusiver Ausschluss (Disjunktion) der spezialisierten Objekttypen: $\bar{a}_1 a_4 \bar{a}_5 \vee \bar{a}_1 a_5 a_4$ mit $a_1 =$ Geschäftspartner, $a_4 =$ natürliche Person (Privatperson) und $a_5 =$ juristische Person (Gesellschaft). Ein Geschäftspartner kann entweder eine natürliche Person oder eine juristische Person, niemals aber beides gleichzeitig sein.

Abstraktes Beispiel:

$S_1 = a_1 a_2 \bar{a}_3, S_2 = a_1 \bar{a}_2 a_3$ (9.1)

Inhaltliche Aussage: Wenn eine Instanzierung der zweckorientierten Datensicht S_1 existiert, darf gleichzeitig keine Instanzierung der zweckorientierten Datensicht S_2 mit der gleichen Ausprägung des Attributes a_1 existieren und umgekehrt. Die folgende Wahrheitstafel zeigt die vollständige Enumeration der Booleschen Va-

blen für die die beiden zweckorientierten Datensichten S_1 und S_2 aus (9.1) und deren Wertzuweisungen $S_1 = f(a_1, a_2, a_3)$ bzw. $S_2 = f(a_1, a_2, a_3)$:

| LfdNr | a_1 | a_2 | a_3 | $S_1 = f(a_1, a_2, a_3)$ | $S_2 = f(a_1, a_2, a_3)$ | $f(a_1, a_2, a_3)$ | |
|-------|-------|-------|-------|--------------------------|--------------------------|--------------------|---|
| 1 | 0 | 0 | 1 | 0*0*1 | 0 | 0*1*0 | 0 |
| 2 | 0 | 0 | 1 | 0*0*0 | 0 | 0*1*1 | 0 |
| 3 | 0 | 1 | 0 | 0*1*1 | 0 | 0*0*0 | 0 |
| 4 | 0 | 1 | 0 | 0*1*0 | 0 | 0*0*1 | 0 |
| 5 | 1 | 0 | 1 | 1*0*1 | 0 | 1*1*0 | 0 |
| 6 | 1 | 0 | 1 | 1*0*0 | 0 | 1*1*1 | 1 |
| 7 | 1 | 1 | 0 | 1*1*1 | 1 | 1*0*0 | 0 |
| 8 | 1 | 1 | 0 | 1*1*0 | 0 | 1*0*1 | 0 |

Tabelle 9: Wahrheitstafel der sich gegenseitig ausschließenden Attribute a_2 und a_3

Wie der Wahrheitstafel zu entnehmen ist, kann eine Instanz zur zweckorientierten Datensicht S_1 genau dann existieren, wenn die Attribute a_1 und a_2 vorhanden sind und gleichzeitig Attribut a_3 nicht vorhanden ist. $S_1 = f(a_1, a_2, a_3) = 1$ in Zeile 7. Ebenso existiert eine Instanz zur zweckorientierten Datensicht S_2 genau dann, wenn die Attribute a_1 und a_3 vorhanden sind und gleichzeitig Attribut a_2 nicht vorhanden ist: $S_2 = f(a_1, a_2, a_3) = 1$ in Zeile 6.

Die weitere Behandlung der Attribute ist wie folgt:

$$a_1 \wedge a_2 \wedge a_3 \vee a_1 \wedge a_2 \wedge a_3 = 1 \tag{9.2}$$

$$a_1 \wedge (a_2 \wedge a_3 \vee \bar{a}_2 \wedge a_3) = 1 \tag{9.3}$$

oder in „Exklusiv Oder Darstellung“:

$$a_1 \wedge (a_2 \times a_3) = 1 \tag{9.4}$$

Das Zeichen „ \times “ steht für „Exklusiv Oder“.

Weist man den Booleschen Variablen (9.3) bzw. (9.4) die Objekttypen $A = a_1$, $B = a_2$ und $C = a_3$ zu, dann erhält man in Objekttypnotation folgenden Ausdruck:

$$A \wedge (B \underline{\vee} \bar{B}C) = 1 \tag{9.5}$$

oder

$$A \wedge (B \times C) = 1 \tag{9.6}$$

Dieser Ausdruck kann graphisch wie folgt interpretiert werden:

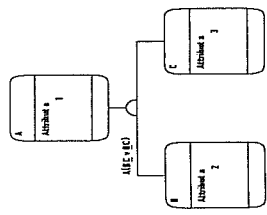


Abbildung 5: Graphische Umsetzung des gegenseitigen Ausschlusses von Objekttypen (disjunkte Objekttypen)

Konkretes Beispiel eines gegenseitigen Ausschlusses von Attributen (Tabelle 4 erweitert):

| Lfd. Nr. | Attribute (Boolesche Variable) | Attribute (verkürzt) | Zweckorientierte Datensichten | | | | | |
|----------|--------------------------------|----------------------|-------------------------------|-----------|---------------------------|--------------|-------|-------|
| | | | Kunde | Lieferant | Gewerblicher Arbeitnehmer | Angestellter | | |
| | 1 | 2 | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 |
| 1 | Ext.PartnerNr | a_0 | + | + | + | + | + | + |
| 2 | KundenNr | a_1 | + | + | + | + | + | + |
| 3 | Name | a_2 | + | + | + | + | + | + |
| 4 | Anschrift | a_3 | + | + | + | + | + | + |
| 5 | Bonität | a_4 | + | + | + | + | + | + |
| 6 | Kreditgrenze | a_5 | + | + | + | + | + | + |
| 7 | LieferantNr | a_6 | + | + | + | + | + | + |
| 8 | PersonalNr | a_7 | + | + | + | + | + | + |
| 9 | Steuersklasse | a_8 | + | + | + | + | + | + |
| 10 | Finanzamt | a_9 | + | + | + | + | + | + |
| 11 | Abteilung | a_{10} | + | + | + | + | + | + |
| 12 | Stundensatz | a_{11} | + | + | + | + | + | + |
| 13 | Lohngruppe | a_{12} | + | + | + | + | + | + |
| 14 | Monatsgehalt | a_{13} | + | + | + | + | + | + |
| 15 | Gehaltsgruppe | a_{14} | + | + | + | + | + | + |

Tabelle 10: Attributabelle mit den zweckorientierten Datensichten „Kunde“, „Lieferant“, „Gewerblicher Arbeitnehmer“ und „Angestellter“ mit sich gegenseitig ausschließenden Attributen

Das „ \times “ - Zeichen in der zweckorientierten Datensicht „Angestellter“ (S_4 , Spalte 6) bedeutet, daß das Attribut Stundensatz a_{11} (Zeile 12) beim einem Angestellten nicht vorkommen darf. Gleiches gilt für das Attribut Lohngruppe a_{12} (Zeile 13). Beide dürfen nur bei der zweckorientierten Datensicht „Gewerblicher Arbeitnehmer“ (S_5 , Spalte 5) vorkommen. Analog dazu dürfen die beiden Attribute Monatsgehalt a_{13} (Zeile 14) und Gehaltsgruppe a_{14} (Zeile 15) nicht in der zweckorientierten Datensicht „Gewerblicher Arbeitnehmer“ (S_5 , Spalte 5) vorkommen. Sie dürfen nur in der zweckorientierten Datensicht „Angestellter“ (S_4 , Spalte 6) vorkommen.

Die disjunktive Normalform aus Tabelle 10 ergibt sich wie folgt:

$$S_1 \vee S_2 \vee S_3 \vee S_4 = \tag{10.0}$$

$$\text{Kunde} \vee \text{Lieferant} \vee \text{Gewerbliche_Arbeitnehmer} \vee \text{Angestellter} = \tag{10.1}$$

$$a_0 \wedge a_1 \wedge a_2 \wedge a_3 \wedge a_4 \wedge a_5 \vee a_0 \wedge a_2 \wedge a_3 \wedge a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \wedge a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14} \vee a_0 \wedge a_2 \wedge a_3 \wedge a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \wedge a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14} = 1 \tag{10.2}$$

Durch Ausklammern von $a_0 \wedge a_2 \wedge a_3$ (10.3) und von $a_7 \wedge a_8 \wedge a_9 \wedge a_{10}$ (10.4) erhält man folgende Ausdrücke:

$$a_0 \wedge a_2 \wedge a_3 \wedge (a_1 \wedge a_4 \wedge a_5 \vee a_6 \vee a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \wedge a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14}) \vee a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \wedge a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14} = 1 \tag{10.3}$$

$$a_0 \wedge a_2 \wedge a_3 \wedge (a_1 \wedge a_4 \wedge a_5 \vee a_6 \vee a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \wedge a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14}) \vee a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14} = 1 \tag{10.4}$$

in „Exklusiv Oder“-Darstellung:

$$a_0 \wedge a_2 \wedge a_3 \wedge (a_1 \wedge a_4 \wedge a_5 \vee a_6 \vee a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \wedge a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14}) \vee a_{11} \wedge a_{12} \wedge a_{13} \wedge a_{14} = 1 \tag{10.5}$$

oder in Objekttypnotation:

$$\text{Externer Partner} = E = a_0 \wedge a_2 \wedge a_3 \tag{10.6}$$

$$\text{Kunde} = K = a_1 \wedge a_4 \wedge a_5 \tag{10.7}$$

$$\text{Lieferant} = L = a_6 \tag{10.8}$$

$$\text{Mitarbeiter} = M = a_7 \wedge a_8 \wedge a_9 \wedge a_{10} \tag{10.9}$$

$$\text{Gewerblicher Arbeitnehmer} = G = a_{11} \wedge a_{12} \tag{10.10}$$

$$\text{Angestellter} = A = a_{13} \wedge a_{14} \tag{10.11}$$

$$E(K \vee L \vee M) \wedge (AG \vee AG) = I \quad (10.12) \text{ oder}$$

$$E(K \vee L \vee M) \wedge (G \cdot A) = I \quad (10.13)$$

Die graphische Interpretation von (10.13) bzw (10.5) ergibt folgende Darstellung

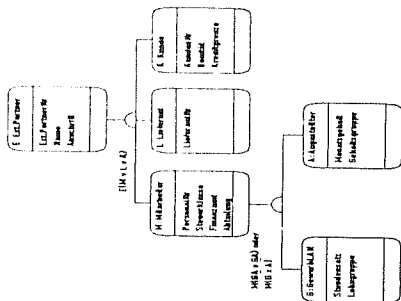


Abbildung 6: Graphische Darstellung des mehrstufigen konkreten und disjunkten Beispiels (10.13) Die beiden Objekttypen G (Gewerblicher Arbeitnehmer) und A (Angestellter) sind zueinander disjunkt. Die Disjunktheit wird durch den Booleschen Ausdruck $M(A \vee AG) \vee M(G \times A)$ ausgedrückt.

1.8 Alternative graphische Darstellung einer Konjunktion

Die graphische Interpretation multiplikativer Verknüpfungen von Variablen außerhalb von Klammern mit den Variablen innerhalb von Klammern durch das Setzen von Spezialisierungssymbolen gilt auch für jede einzelne Konjunktion. Eine Konjunktion, die graphisch alternativ dargestellt werden soll, ist bereits generalisiert / spezialisiert und ihre Attribute (Boolesche Variable) z.B. a_1, a_2, a_3 sind zunächst einem Objekttyp (z.B. A) zugewiesen. Die einzelnen Variablen einer Konjunktion sind nichts anderes als multiplikative Verknüpfungen der Form: $a_1, a_2, a_3 = a_1(a_2, a_3) = a_1(a_2, a_3) = \dots$ u.s.w.. Sie können für bestimmte Zwecke graphisch wie Disjunktionen interpretiert werden, mit dem Unterschied, daß dann spezialisierte Objekttypen der gleichen Ebene untereinander nicht disjunktiv sondern konjunktiv verbunden sind.

In den folgenden Darstellungen werden beispielhaft die drei Variablen a_1, a_2 und a_3 kombiniert bzw. variiert. Die Ausdrücke sind identisch. Dies gilt auch für die daraus abgeleiteten graphischen Interpretationen. Jede Konjunktion kann auf diese Weise dargestellt werden. Damit besitzt der Entwickler die Möglichkeit, bei Bedarf jederzeit konjunktive Strukturen graphisch in einfachere Terme (wieder Konjunktionen) zu zerlegen und in das Datenmodell zu integrieren. Die Darstellungen sind bezüglich der Möglichkeiten, drei Variablen zu kombinieren bzw. zu variieren, nur beispielhaft und deshalb nicht vollständig. Die drei Booleschen Variablen a_1, a_2, a_3 können wie folgt einfacheren Objekttypen zugewiesen werden:

| LfdNr | Attributvarian | Objekttyp- / Attributzusordnung | Objekttypenvarian n |
|-------|-------------------|---------------------------------|------------------------|
| 1 | a_1, a_2, a_3 | $A' = a_1, a_2, a_3$ | A' |
| 2 | $(a_1, a_2), a_3$ | $A = a_1, B = a_2, a_3$ | A(B) |
| 3 | $a_1, (a_2, a_3)$ | $A = a_1, a_3, B = a_2$ | A(B) |
| 4 | a_1, a_2, a_3 | $A = a_1, B = a_2, C = a_3$ | A(B,C) |
| 5 | $a_1, (a_2, a_3)$ | $A = a_1, B = a_2, C = a_3$ | A(B)(C) |
| 6 | a_1, a_2, a_3 | $A = a_1, B = a_2, C = a_3$ | A(B) wie 2 |
| 7 | ... | ... | ... |

Die graphischen Interpretationen zu Tabelle 11 sind wie folgt:

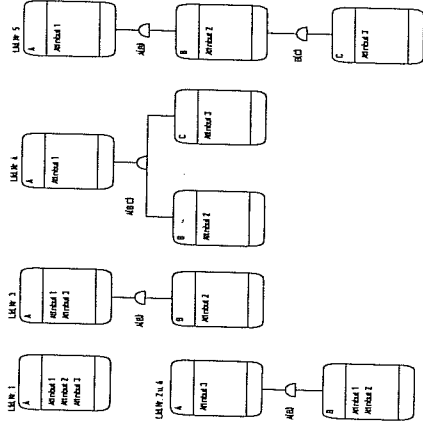


Abbildung 7: Beispiele der alternativen graphischen Darstellung der Konjunktion $A' = a_1 a_2 a_3$

Die Elemente der Spezialisierung (untergeordnete Ebene) sind mit dem Element der Generalisierung (übergeordnete Ebene) konjunktiv (multiplikativ) verknüpft (vertikale Verknüpfung), z.B. A(B,C). Die Elemente der Spezialisierung (der untergeordneten Ebene) sind untereinander ebenfalls konjunktiv verknüpft (horizontale Verknüpfung), z.B. A(B,C). Es liegt hier der Spezialfall der Disjunktion mit $B = 1$ und $C = 1$ vor: $A(B \vee C) = A(B,C)$. Dies gilt über alle Ebenen eines mehrstufigen Systems hinweg.

Die multiplikative Verknüpfung erzwingt, daß die jeweiligen Attribute bzw. die sie vertretenden Objekttypen jeweils alle oder gar nicht vorhanden sind: Ist ein Attribut nicht vorhanden (nicht belegt, =0), dann wird das gesamte Produkt (die Konjunktion) zu Null, d.h. diese Ausprägung des Objekttyps und alle konjunktiv mit diesem Objekttyp verbundenen Objekttypen gibt es nicht.

Graphisch ist die Potenzmenge aller Attribute ohne die leere Menge in einer Konjunktion zur getrennten Darstellung als Objekttypen zugelassen. Es sollte jedoch ein Minimum an Spezialisierungen (an spezialisierten Objekttypen) angestrebt werden.

Eine Menge dieser Potenzmenge kann jede mögliche Variation annehmen. Alle Darstellungsformen sind als inhaltlich identisch anzusehen. Sie genügen dem Kommutativgesetz der Booleschen Algebra für die Logische Multiplikation.

2 Hauptgeneralisierung / -spezialisierung

2.1 Bestimmung der Wurzelobjekttypen bei Generalisierungen / Spezialisierungen

Erweitert man die Tabelle 10 um die zweckorientierte Datensicht „Parzähler“ und um die Attribute Bankleitzahl, Bankkonto-Nr., Debitoren-Nr. und Kreditoren-Nr., dann ergibt sich folgende Tabelle mit den jeweiligen Zuweisungen dieser Attribute zu den zweckorientierten Datensichten. Gleichzeitig wurde der gegenseitige Ausschluß von Attributen der zweckorientierten Datensichten „Lieferant“ und „Gewerblicher Arbeitnehmer“ bzw. „Angestellter“ vorgenommen. Dies bedeutet, daß eine bestimmte Instanz eines „Externen Partners“ nicht gleichzeitig ein „Lieferant“ und ein „Gewerblicher Arbeitnehmer“ bzw. „Angestellter“ sein kann und umgekehrt. Bestimmt man die Mächtigkeit der Attribute über alle zweckorientierten Datensichten (systemweit) hinweg, dann erhält man die am häufigsten vorkommenden Attribute (Spalte 8):

(11.19)

$$E (B (K \cdot R \vee D (L \times M (G \times A))) \vee K) = 1$$

Dieser Ausdruck läßt sich wie folgt graphisch darstellen:

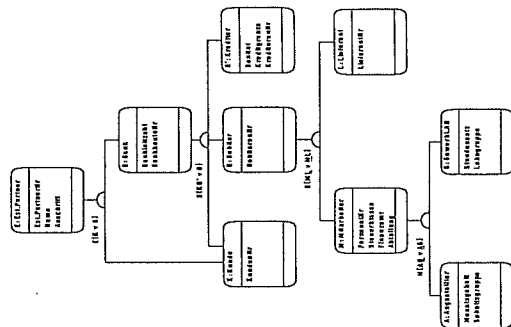


Abbildung 8: Vernetzte graphische Darstellung des Ausdrucks (11.19)

Diese Mehrfachvererbung ist intuitiv vorgekommen worden, in dem die in Punkt 1.8 beschriebene Vorgehensweise benutzt wurde. Die beiden Objekttypen KR sind multiplikativ miteinander verknüpft und können, falls sie vorkommen, immer nur gemeinsam vorkommen.

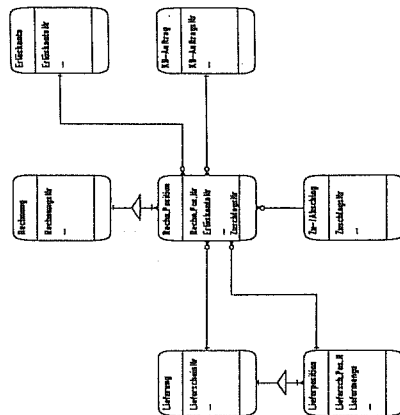


Abbildung 9: Datenmodellausschnitt der Kundenrechnungsposition und deren „referentielle Umgebung“

Ein weiteres Beispiel soll den Sachverhalt nochmals verdeutlichen. Abbildung 9 zeigt einen Ausschnitt aus dem Datenmodell eines Güterunternehmens. Es zeigt den verketteten Entitytyp [SchNe90] Rechnungsposition und seine „referentielle Umgebung“. Die Rechnungsposition ist grau hinterlegt und lokal noch nicht generali-

siert / spezialisiert. Sie stellt nur einen zusammengefaßten Objekttyp dar, der durch den Generalisierungs- / Spezialisierungsprozeß noch zu verfeinern ist. Das Datenmodell wurde mit SQL realisiert.

Gegeben waren folgende Attribute mit den dazugehörigen zweckorientierten Datensichten aus der Kundenrechnungsposition. Die Attribute sind nur stellvertretend ausgewählt worden und deshalb nicht vollzählig.

| Lfd. Nr. | Attribute (Boolesche Variable) | Zweckorientierte Datensichten der Rechnungsposition | | | | | Mächt.-Attribut |
|----------|--------------------------------|---|-----------------|--------------------------|------------------|-----------------|-----------------|
| | | Artikel-zuschlag | Liefer-zuschlag | Kunden-auftrags-zuschlag | Artikel-zuschlag | Mächt.-Attribut | |
| 1 | Rechnungsposition-Nr. | S ₁ | S ₂ | S ₃ | S ₄ | 7 | |
| 2 | Erlöskonto-Nr. | a ₁ | a ₂ | a ₃ | a ₄ | 4 | |
| 3 | Kundenauftrags-Nr. | a ₅ | a ₆ | a ₇ | a ₈ | 4 | |
| 4 | Lieferschein-Nr. | a ₉ | a ₁₀ | a ₁₁ | a ₁₂ | 3 | |
| 5 | Lieferscheinposition-Nr. | a ₁₃ | a ₁₄ | a ₁₅ | a ₁₆ | 2 | |
| 6 | Zuschlags-Nr. | a ₁₇ | a ₁₈ | a ₁₉ | a ₂₀ | 3 | |
| 7 | Berechnungsmenge | a ₂₁ | a ₂₂ | a ₂₃ | a ₂₄ | 2 | |
| 8 | Einzelpreis je Mengeneinh. | a ₂₅ | a ₂₆ | a ₂₇ | a ₂₈ | 2 | |
| 9 | Zuschlagsverrechn. Kennz. | a ₂₉ | a ₃₀ | a ₃₁ | a ₃₂ | 1 | |
| 10 | Zuschlagsbetrag einmalig | a ₃₃ | a ₃₄ | a ₃₅ | a ₃₆ | 2 | |

Tabelle 13: Attributabelle „Rechnungsposition“ mit den zweckorientierten Datensichten „Artikel“, „Artikelzuschlag“, „Lieferzuschlag“ und „Kundenauftragszuschlag“

Aus Vereinfachungsgründen wurde in dieser Abhandlung auf eine Darstellung verzichtet, in der alle nicht belegten Attribute sich gegenseitig ausschließen.

Aus dieser Tabelle läßt sich folgende disjunktive Normalform ableiten:

$$S_1 \vee S_2 \vee S_3 \vee S_4 = a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_1 a_2 a_3 a_4 a_5 a_6 a_7 a_8 a_9 = 1 \quad (12.1)$$

In diesem Ausdruck kommen die beiden Booleschen Variablen a₁ und a₂ viermal vor. Sie sind deshalb zunächst auszuklammern:

$$a_1 a_2 (a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_4 a_5 a_6 a_7 a_8 a_9) = 1 \quad (12.2)$$

Danach könnte entweder die Boolesche Variable a₄ oder die Variable a₆ ausgeklammert werden. Beide kommen dreimal vor. Zunächst soll einmal die Variable a₄ ausgeklammert werden:

$$a_1 a_2 (a_4 (a_3 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_5 a_6 a_7 a_8 a_9 \vee a_3 a_5 a_6 a_7 a_8 a_9) \vee a_3 a_5 a_6 a_7 a_8 a_9) = 1 \quad (12.3)$$

Die Variable a₆ ließe sich jetzt nur noch in der inneren Klammer ausklammern. In dieser Vorgehensweise ist dies nicht erlaubt, da sich dieselbe Variable nochmals in einer anderen Klammer befindet. Damit besteht nur noch die Möglichkeit, in der inneren Klammer die Variablen a₃, a₇ und a₈ auszuklamern:

$$a_1 a_2 (a_4 (a_3 a_7 a_8 (1 \vee a_5 a_6) \vee a_5 a_6 a_7 a_8) \vee a_3 a_5 a_6 a_7 a_8) = 1 \quad (12.4)$$

Eine Zuweisung der Attribute zu Objekttypen ergibt:

$$A = a_1 a_2, D = a_4, L = a_5 a_6 a_7 a_8, M = a_3 a_5 a_6 a_7 a_8, K = a_3 a_5 a_6 a_7 a_8 a_9$$

Der Ausdruck (12.4) läßt sich damit auch wie folgt darstellen:

$$A (D ((1 \vee L) \vee K) \vee M) = 1 \quad (12.5)$$

Folgende Darstellung zeigt diesen Ausdruck in graphischer Form, wobei aber die mehrfach vorkommenden Attribute a_6 (Zuschlagsnummer) und a_{10} (Zuschlagsbetrag einmally) in den Objekttypen L, K und M hingegen angenommen werden müßten. Da das Attribut a_6 (Zuschlagsnummer) in den Objekttypen K, L und M vorkommt, ist für jeden Objekttyp eine Beziehung zum Entitytyp [SciNé90] ZU- / ABSCHLAG vorhanden:

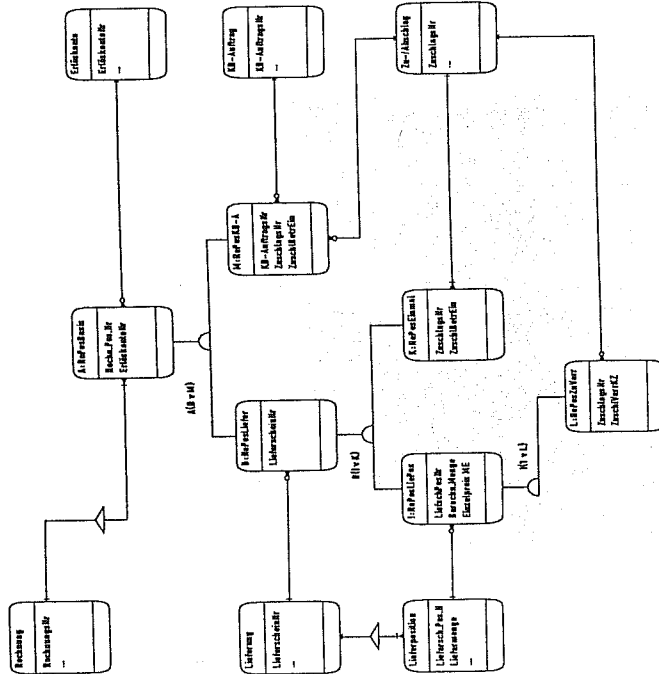


Abbildung 10: Graphische Darstellung der Rechnungsposition Ausdruck (12.5)
Zerlegt man die Objekttypen $L = a_6 a_9$, $K = a_6 a_{10}$ und $M = a_6 a_9 a_{10}$ nochmals nach den Regeln der alternativen graphischen Darstellung einer Konjunktion (Punkt 1.8), dann kann eine Vernetzung abgeleitet werden:

$$L = (a_6)(a_9), K = (a_6)(a_{10}) \text{ und } M = (a_6)(a_9)(a_{10}).$$

L wird ersetzt durch $B = a_6$ und $H = a_9$, K wird ersetzt durch $B = a_6$ und $C = a_{10}$ und M wird ersetzt durch $E = a_6, B = a_9$ und $C = a_{10}$.

Der Ausdruck (12.5) ändert sich damit in:

$$A(D((I \vee BH) \vee BC) \vee BCE) = 1 \quad (12.5.1)$$

Daraus kann die folgende vernetzte graphische Darstellung abgeleitet werden. Wie zu erkennen ist, besteht zwischen dem Entitytyp ZU- / ABSCHLAG jetzt nur noch eine Beziehung zum spezialisierten Objekttyp B über das Attribut a_6 (Zuschlagsnummer). Dieses Attribut wird den anderen generalisierten / spezialisierten Objekttypen vererbt:

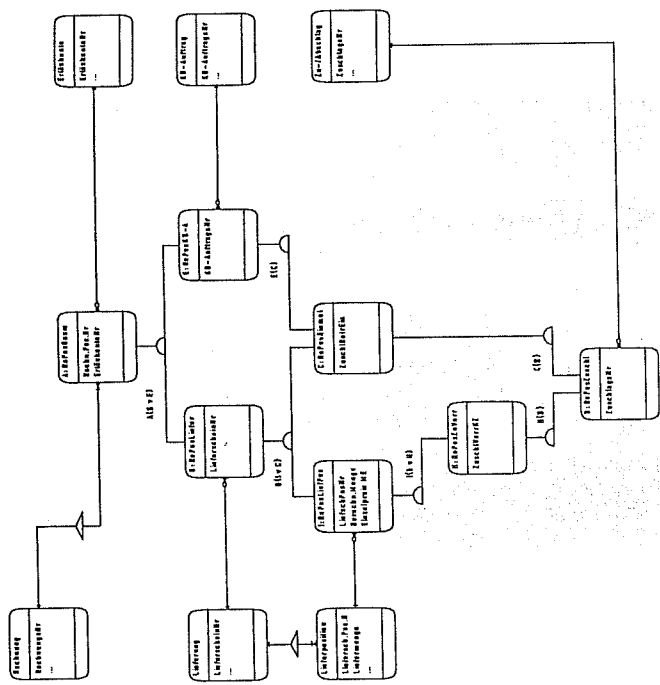


Abbildung 11: Vernetzte graphische Darstellung der Rechnungsposition Ausdruck (12.5) mit Mehrfachvererbung
Gegenüber der Darstellung in Abbildung 10 sind die referentiellen Beziehungen aus den Umgebungsobjekttypen jetzt genau den generalisierten / spezialisierten Objekttypen zuzuordnen, in denen sich die entsprechenden Fremdschlüssel befinden.

2.3 Elementare Attribute (Variable)

Boolesche Variable, die nicht vor Klammern gezogen wurden, gelten als elementar. In (12.4) sind dies die Variablen a_1, a_6, a_9 und a_{10} . Sie bilden die „Blätter“ in der baumartigen Hierarchie. Der Ausdruck (12.4) ist das Ergebnis der Hauptgeneralisierung / -spezialisierung. In der Nebengeneralisierung / -spezialisierung können die beiden in Ausdruck (12.4) redundant vorkommenden Variablen a_6 und a_{10} weiter behandelt werden, damit nur sie einmal vorkommen aber mehrfach vererbt werden können. Die Nebengeneralisierung / -spezialisierung kann in einfachen Fällen intuitiv vorgenommen werden, indem die zu vernetzenden Terme mit Hilfe der in Punkt 1.8 beschriebenen alternativen Darstellung konjunktiv verbundener Attribute zerlegt werden. Eine formale Vorgehensweise zur Nebengeneralisierung / -spezialisierung, die zur Mehrfachvererbung führen, wird in Punkt 5 dieser Abhandlung beschrieben.

Mehrfach vorkommende Boolesche Variable können in verschiedenen Ebenen einer Hauptgeneralisierung / Spezialisierung vorkommen. Zur weiteren Behandlung in einer Nebengeneralisierung / -spezialisierung dürfen sie nur in den „Blättern“ und nicht in den „Zweigen“ (Knoten) des Baumes einer Hauptgeneralisierung / -spezialisierung vorkommen. Falls sie in Knoten des Generalisierungs / -spezialisierungsbaumes vorkommen dann ist ihre Vernetzung mehrdeutig. Damit kann die Kontrolle der referentiellen Integrität über zwei benachbarte Ebenen nicht mehr durch einen Booleschen Ausdruck gewährleistet werden, in dem sich nur Objekttypen aus den zwei direkt benachbarten Ebenen befinden.

Um mehrdeutige Knoten in vernetzten Datenmodellen der Hauptgeneralisierung / -spezialisierung zu vermeiden, ist folgende Regel zu beachten:

Kommt eine Boolesche Variable oft in verschiedenen Stufen eines Ausdrucks der Hauptgeneralisierung / -spezialisierung vor, dann darf sie lokal nicht ausgeklammert werden. Eine mehrfach vorkommende Boolesche Variable muß elementar bleiben. Damit wird sichergestellt, daß mehrfach vorkommende Boolesche Variable sich nur in den „Blättern“ einer baumartigen Hierarchie der Hauptgeneralisierung / -spezialisierung befinden. Sie sind in einer Nebengeneralisierung / -spezialisierung weiter zu behandeln. Auf den Ausdruck (12.3) bezogen bedeutet dies: Die in der inneren Klammer zweimal vorkommende Boolesche Variable a_6 darf nicht ausgeklammert werden, da sie nochmals in der äußeren Klammer (letzter Term $a_9 \wedge a_{10}$) vorkommt.

2.4 Mehrere gültige Lösungen bei Generalisierungen / Spezialisierungen
 Wird in Ausdruck (12.2) nicht die Variable a_4 sondern die Variable a_6 ausgeklammert, was formal völlig gleichwertig ist, dann ergibt sich folgender Ausdruck:

$$a_1 \wedge a_2 \wedge (a_4 \wedge a_5 \wedge a_7 \wedge a_8 \vee a_6 \wedge a_4 \wedge a_{10} \vee a_3 \wedge a_{10}) = 1 \tag{12.6}$$

Danach ist die Variable a_{10} in der inneren Klammer auszuklammern. Die Variable a_4 kann nicht ausgeklammert werden, da sie in einer anderen Klammernebene nochmals elementar vorkommt:

$$a_1 \wedge a_2 \wedge (a_4 \wedge a_5 \wedge a_7 \wedge a_8 \vee a_6 \wedge (a_4 \wedge a_5 \wedge a_7 \wedge a_8 \vee a_{10} \wedge (a_4 \wedge a_3))) = 1 \tag{12.7}$$

Eine Zuweisung der Attribute aus Ausdruck (12.7) zu Objekttypen ergibt:

$$A = a_1, a_2, G = a_4, a_5, a_7, a_8, B = a_6, F = a_4, a_5, a_7, a_8, a_9, C = a_{10}, D = a_4, E = a_3.$$

Der Ausdruck (12.7) läßt sich damit auch wie folgt darstellen:

$$A \wedge (G \vee B \wedge (F \vee C \wedge (D \vee E))) = 1 \tag{12.8}$$

Folgende Darstellung zeigt diesen Ausdruck in graphischer Form, wobei hier die mehrfach vorkommenden Attribute a_4, a_5, a_7 und a_8 in den Objekttypen G, F und D hingenommen werden mußten:

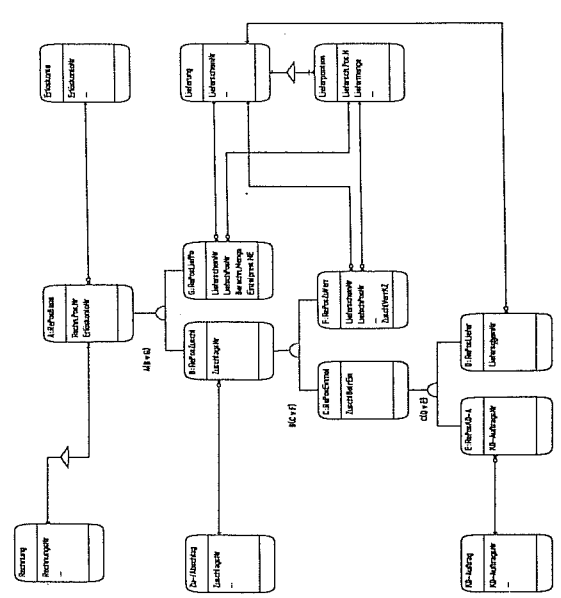


Abbildung 12: Graphische Darstellung Rechnungsposition Ausdruck (12.8)

Zerlegt man die Objekttypen $G = a_4 \wedge a_5 \wedge a_7 \wedge a_8$, $F = a_4 \wedge a_5 \wedge a_7 \wedge a_8 \wedge a_9$ und $D = a_4$ nochmals nach den Regeln einer alternativen graphischen Darstellung einer Konjunktion (Punkt 1.8), dann kann eine Vernetzung abgeleitet werden:

$$G = (a_4) \wedge (a_5 \wedge a_7 \wedge a_8), F = (a_4) \wedge (a_5 \wedge a_7 \wedge a_8) \wedge (a_9), D = (a_4).$$

G wird ersetzt durch $D = (a_4)$ und $I = (a_5 \wedge a_7 \wedge a_8)$ und F wird ersetzt durch $D = (a_4)$, $I = (a_5 \wedge a_7 \wedge a_8)$ und $H = (a_9)$.

Damit wird der Ausdruck (12.8) durch folgenden Ausdruck ersetzt:

$$A \wedge (D \vee B \wedge (D \vee H \vee C \wedge (D \vee E))) = 1 \tag{12.9}$$

Daraus kann die folgende vernetzte graphische Darstellung abgeleitet werden:

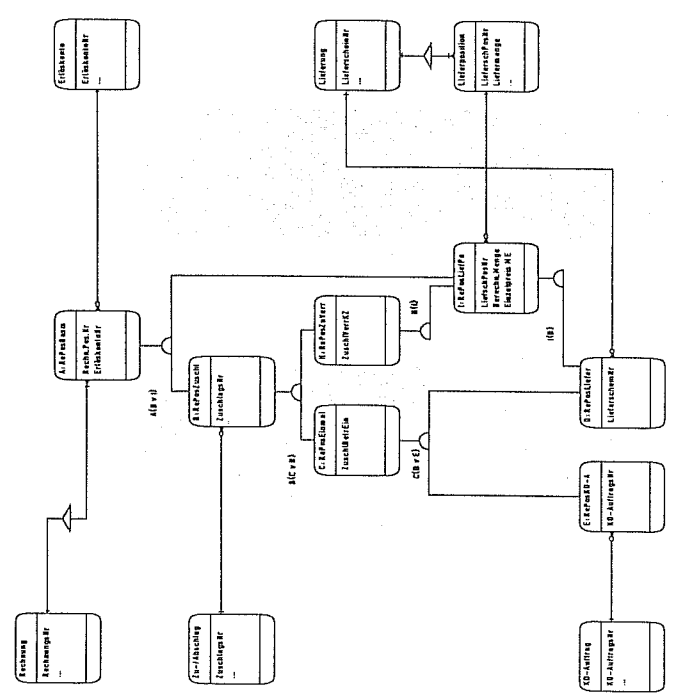


Abbildung 13: Vernetzte graphische Darstellung der Rechnungsposition Ausdrucks (12.8) mit Mehrfachvererbung

Die beiden Ausdrücke (12.4) und (12.7) sind semantisch völlig gleichwertig. Die Anzahl der Objekttypen ist gleich. Wie jedoch die beiden graphischen Darstellungen zeigen, hat die Darstellung der Abbildung 13 eine Generalisierung / Spezialisierung weniger als die Darstellung der Abbildung 11. Deshalb wäre der Ausdruck (12.7) vorzuziehen, da er bei gleichem Inhalt weniger Elemente zur Repräsentation benötigt.

Eine weitere Entscheidungsmöglichkeit, wenn mehrere gültige Lösungen bei Generalisierungen / Spezialisierungen vorliegen, wäre die Auswahl der Alternative, bei der die häufigsten Instanzierungen zur Generalisierungs- / Spezialisierungswurzel hin zu erwarten sind. Damit werden immer immer minimale Instanzentzweige bzw. -bäume ausgeprägt. Dies hätte zur Folge, daß der Zeitverbrauch für Verarbeitungsprozesse insgesamt ein Minimum annehmen würde. Dasselbe gilt für die Anzahl der dann tatsächlich ausgeprägten Instanzen.

Die Vorgehensweise stellt sicher, daß eine Instanziierung von generalisierten / spezialisierten Objekttypen immer von der jeweiligen Generalisierungswurzel aus lückenlos bis zum jeweiligen letzten spezialisierten Objekttypen (Blatt) erfolgt. Dies ist z.B. in der objektorientierten Systementwicklung Voraussetzung für die Vererbung von Attributen und Operationen (Methoden, Services). Für jeden direkten generalisierten / spezialisierten Objekttypnachbar ist die referentielle Integrität durch einen Boolesch algebraischen Ausdruck gesichert.

- 3 Instanziierungstypen (in Vorbereitung)
- 4 Systematisierte und integrierte Datenmodell- und Instanzenwartung (in Vorbereitung)
- 5 Nebengeneralisierung / -spezialisierung (in Vorbereitung)

Literaturverzeichnis:

- [Bart77] Bartsch, H.-J., Taschenbuch Mathematischer Formeln, Leipzig, 1977
- [CoYo90] Coad, P., Yourdon, E., Object-Oriented Analysis, Englewood Cliffs, New Jersey, 1990
- [FeSi93] Fersli, O. K., Sinz, E. J., Grundlagen der Wirtschaftsinformatik, Band 1, München, 1993
- [Hohe93] Hohenstein, U., Formale Semantik eines erweiterten Entity-Relationship-Modells, Leipzig, 1993
- [InDu89] Informatik Duden, Mannheim, 1989
- [Mar88] Marty, R., Von der Subroutineintechnik zu Klassenhierarchien - Eine schrittweise Hinführung zu objektorientierter Programmierung, Zürich-Irchel, 1988
- [Sche90] Scheer, A.-W., Wirtschaftsinformatik, Heidelberg, 1990
- [Sché90] Schönhaler, F., Németh, T., Software-Entwicklungswerkzeuge: Methodische Grundlagen, Stuttgart, 1990

Die gesamte Vorgehensweise wird etwa Mitte Dezember 1995 als Buch im VDE-Verlag, Offenbach, unter dem Titel „Objektorientierte Analyse - mit Hilfe der Booleschen Algebra“, Autor Gisbert Englmeier, erscheinen.

Natürlichsprachlicher Entwurf von Informationssystemen

Jan 2

Erich Ortner
Universität Konstanz, Informationswissenschaft
Lehrstuhl für Informationsmanagement
Postfach 5560
78434 Konstanz
Tel.: 07531-88-2976
Fax: 07531-88-2601
e-mail: ortner@inf-wiss.uni-konstanz.de

In dem Beitrag werden verschiedene linguistische Methoden der Informationssystementwicklung vorgestellt. Dabei wird der Versuch unternommen, sie nach einem einheitlichen Schema (Tabelle 1) zu klassifizieren. Zum Schluß wird auf einen Workshop zum "Natürlichsprachlichen Entwurf von Informationssystemen" hingewiesen, der vom 28.05. - 31.05.1996 in Tutzing stattfinden wird.

1. Einleitung

Linguistische Methoden der Informationssystem-Entwicklung sind dadurch charakterisiert, daß mit ihnen - meist syntaktisch und semantisch reglementierte - natürliche Sprachen (z. B. Englisch, Japanisch, Deutsch) bei der Entwicklung von Informationssystemen methodisch eingesetzt werden. Sie kommen komplementär zu semiformalen Diagrammethode - z. B. aus dem Requirements Engineering, der Wissensrepräsentation oder aus der Datenbank-Anwendungsentwicklung - in der Anfangsphase einer Systementwicklung sowie bei der Einführung und Nutzung der Systeme zum Einsatz. Ein besonderes Kennzeichen ist dabei, daß ein Entwurfsergebnis mit ihnen nicht

wie beispielsweise bei Diagrammmethoden aus "Knoten" und "Kanten", sondern aus "Begriffen" (Fach- oder Themenwörtern), die zu "Aussagen" (Sätzen) verbunden werden, entwickelt wird. Die ersten Entwicklungsergebnisse sind Aussagensammlungen, aus denen alle weiteren Ergebnisse einer Informationssystemlösung - z. T. automatisiert - abgeleitet werden können.

Diese Entwicklung setzte Anfang der 80er Jahre ein, weil mit Konzepten wie "rapid prototyping", dem Einsatz "graphischer Entwurfsmethoden" oder einem unter "demokratischen Vorzeichen" praktizierten (benutzer-)partizipativen Entwicklungsstil entscheidende Qualitätsverbesserungen bei der Erstellung von Informationssystemen nicht mehr erzielt werden konnten. Vor allem die fachliche Software-Qualität (z. B. Konsistenz der Begriffsbildung bei Benutzern und den implementierten Systemen) läßt sich mit diesen Ansätzen nicht wesentlich verbessern, und der ingenieurmäßige Arbeitsprozeß - methodisches, auf Ideen und verantwortbaren Entscheidungen basiertes, rationelles Problemlösen - wird bei manchen Ansätzen teilweise entstellt dargestellt.

Neuere Ansätze tendieren deshalb dazu, mehr von dem "natürlichen" Sprach-einsatz der Benutzer durch linguistische Entwurfsmethoden in den Entwicklungs- und Nutzungsprozeß der Anwendungssoftware zu integrieren. Man geht dabei von der Feststellung aus, daß Menschen ihre Arbeitsprozesse (Business Process Engineering, Workflow-Modellierung) mit Fachsprachen organisieren. Desgleichen werden Fachsprachen zur Informationsverarbeitung und Kommunikation bei der Aufgabenerledigung in einer Organisation eingesetzt. Damit ist eine (natürliche) Sprache, wenn sie mit ihrer Grammatik (Satzbildungsregeln) und ihren relevanten Fachbegriffen (Fachtermini) aus den Anwendungsbereichen exakt festgelegt ist, ein "ideales" Konstruktionsinstrument sowohl für Organisationsstrukturen (Aufbau- und Ablauforganisation) als auch für Informationssysteme (z. B. Anwendungssystementwicklung) eines Unternehmens.

Chomsky [11] hat die Bedeutung einer "natürlichen Sprache" für Menschen einmal so charakterisiert: "Der Mensch lernt nicht eigentlich eine Sprache, sondern er lernt in einer Sprache schöpferisch tätig zu sein." Diese Einsicht scheint bei linguistischen Methoden der Informationssystementwicklung ein Leitmotiv ihres Einsatzes zu bilden.

2. Arbeiten zu linguistischen Methoden der Informationssystementwicklung

Die verschiedenen linguistischen Ansätze in Tabelle 1 können zunächst durch die Beantwortung der Frage eingeteilt werden, ob eine existierende (natürliche) Sprache zur Entwicklung der Informationssysteme eingesetzt wird, oder ob eine "quasi-natürliche" Sprache zu diesem Zweck gemeinsam mit Benutzern neu aufgebaut wird. Linguistische Entwicklungsmethoden vom ersten Typ werden empirische Ansätze und linguistische Methoden vom zweiten Typ konstruktive Ansätze genannt. Bei den empirischen Ansätzen können empirisch-analytische und empirisch-experimentelle Methoden unterschieden werden.

Empirisch-analytische Methoden stellen im Entwicklungsprozeß natürlich-sprachliche Aussagen der Benutzer über Sachverhalte eines Anwendungsgebiets mit künstlichen formalen Mitteln (z. B. mit mathematischen Sprachen, Diagrammsprachen oder einer Logiksprache) "exakt" dar. In empirisch-experimentellen Projekten werden Kommunikationsverhältnisse, wie sie sich z. B. im Bürobereich ergeben, mittels protokollierter Sprechakte simuliert. Diese Ansätze wurden auf Basis von Sprechakttheorien - z.B. in Anlehnung an Austin [5] oder Searle [41] - entwickelt.

Bei den empirisch-analytischen Ansätzen werden die entwickelten Systeme aus der Intension der (Benutzer-)Fachbegriffe "interpretativ" erarbeitet. Bei empirisch-experimentellen Ansätzen kommt man primär über die Extension der (Benutzer-)Fachbegriffe zu einem Systementwurf.

Als Intension bezeichnet man seit Carnap [9] die Bedeutung oder auch die Merkmale bzw. die Definition eines Begriffs. In der Datenmodellierung werden z. B. Attribute, die zu einem Objekttyp (Fachbegriff) zusammengefaßt werden, die Intension eines Objekttyps [45] genannt. Unter der Extension eines Begriffs faßt man alle Objekte, für die ein Begriff zutrifft (die unter den Begriff fallen), zusammen. In der Datenmodellierung werden die Ausprägungen eines Objekttyps - also die Beschreibungen existierender (realer) Objekte (Tupel, Datensätze) - seine Extension [45] genannt.

Konstruktive linguistische Methoden (Tabelle 1) sind nicht empirisch (im Sinne von "aufdecken"), sondern normativ (im Sinne von "herstellen") definiert.

| Position Gebiet | empirisch | | konstruktiv |
|--|--|--|--|
| | analytisch | experimentell | |
| Datenbank- Anwendungs- entwicklung | <ul style="list-style-type: none"> - Chen '83 [10] - Colombetti et al. '85 [12] - Buitelaar/ van de Riet '92 [8] - Métails et al. '93 [31] - Tjoa/Berger '93 [43] | <ul style="list-style-type: none"> - Albrecht et al. '95 [2] | <ul style="list-style-type: none"> - Wedekind '81 [46] - Ortner '82 [35] - Ortner/Söllner '89 [36] - Vogler '94 [44] |
| Wissens- repräsen- tation | <ul style="list-style-type: none"> - Weigand '90 [48] - Dignum/van de Riet '91 [13] - Gerstl '92 [17] | | <ul style="list-style-type: none"> - Gunia '94 [18] |
| Objektorien- tierter Ent- wurf | <ul style="list-style-type: none"> - Abbott '83 [1] - Saeki et al. '89 [39] - Kristen '94 [24] - Mich/Garigliano '95 [32] | | <ul style="list-style-type: none"> - Wedekind '92 [47] - Schienmann '95 [40] |
| Büroinforma- tionssysteme (Groupware, Workflow- Systeme) | | <ul style="list-style-type: none"> - Lehtinen/ Lyyti- nen '86 [27] - Winograd '87/88 [50] - Flores et al. '88 [16] - Auramäki et al. '88 [4] | |
| Informations- systement- wicklung (generell) | <ul style="list-style-type: none"> - Wintraecken '89 [51] - Yonezaki '89 [52] - Rolland/ Proix '92 [38] - Lacity/ Janson '94 [25] | <ul style="list-style-type: none"> - Andersen '91 [3] - Holmquist/ Andersen '91 [22] | <ul style="list-style-type: none"> - Müller-Merbach '83 [33] - Ortner '94 [37] |

Tabelle 1: Arbeiten zu linguistischen Methoden der Informationssystem-
entwicklung

Bei konstruktiven Ansätzen wird eine reglementierte Sprache zur Informationssystementwicklung zusätzlich hergestellt, während bei empirischen Ansätzen eine vorhandene Sprache im Entwicklungsprozeß (mit formalen Mitteln) "aufgedeckt" wird. Bei konstruktiven Ansätzen besteht ein Ziel oft darin, zwecks besserer Entwicklung und Nutzung der Informationssysteme, die faktische Sprachgenese in den Anwendungsbereichen um eine rekonstruierte normative Genese der Benutzerfachsprachen zu ergänzen.

Bezüglich einer Einteilung der Arbeiten zum natürlichsprachlichen Entwurf von Informationssystemen in Tabelle 1 kann man feststellen, daß empirische Ansätze im allgemeinen eine formale Sprache (Grammatik) und konstruktive Ansätze eine materiale Sprache (Grammatik und Fachlexikon) zur Entwicklung von Informationssystemen einsetzen [37].

2.1. Empirisch-analytische Methoden der Informationssystementwicklung

Empirisch-analytische Entwicklungsmethoden basieren i. d. R. auf einem formalen Sprachansatz. Neben einer (formalen) Objektsprache - z. B. einer Diagrammsprache -, die als Entwicklungssprache eingesetzt wird, ist eine weitere (materiale) Metasprache - in Form einer Benutzerfachsprache - erforderlich, in der die "Variablen" in den objektsprachlichen Konstrukten (z. B. Rechtecke, Pfeile, Dreiecke) ggf. gemeinsam mit Benutzern fachsprachlich interpretiert werden. Erst durch Interpretation ergeben sich aus konstruierten Aussageformen oder graphischen Konstrukten "inhaltliche" Aussagen.

Bei konstruktiven Entwicklungsmethoden werden im Vergleich dazu öfter materiale Entwicklungssprachen eingesetzt, die aus einer Grammatik (formaler Teil) **und** aus einem mit rekonstruierten Fachbegriffen "gefüllten" Lexikon (materialer Teil) bestehen. Eine Unterscheidung zwischen Objekt- und Metasprachen ist bei ihrem Einsatz nicht erforderlich.

In der Informatik als Ingenieurdisziplin überwiegen natürlich die formal-sprachlichen Ansätze, während in traditionellen Ingenieurdisziplinen - z. B. im Maschinenbau - materiale Sprachen zum Einsatz kommen. Hier werden "darstellende Geometrie" und "technisches Zeichnen" als formaler Teil

(Grammatik) einer Konstruktionsprache und die Teildisziplin "Maschinenelemente" als ihr materialer Teil (Lexikon) angesehen.

Einem formalsprachlichen Ansatz folgend schlägt z. B. Chen [10] zur besseren Verständigung zwischen Entwicklern und Benutzern vor, den Informationsbedarf einer Anwendung zunächst in Englisch zu dokumentieren. Die englischen Beschreibungen werden dann von Datenbankdesignern in Entity-Relationship-Diagramme (oder ähnliche formale Repräsentationssprachen) "umgewandelt". In diesem Beitrag [10] untersucht Chen die (syntaktische) Übereinstimmung zwischen englischen Satzstrukturen und Entity-Relationship-Diagrammen und schlägt elf Regeln für die "Übersetzung" vor. Die Regeln für die Ermittlung (Rekonstruktion) der Fachbegriffe bleiben dabei allerdings im Dunkeln. Eine fachbegrifflich unreflektierte natürlichsprachliche Repräsentation wird in eine fachbegrifflich unreflektierte diagrammsprachliche Repräsentation transformiert. Die grundlegenden Konstrukte des Englischen - Substantiv, Verb, Adjektiv, Adverb, Gerundium, Satz - finden Gegenstücke in der Entity-Relationship-Diagrammtechnik.

Beim Entwurf von Informationssystemen werden Wörter (Fachtermini) mit einer festen Bedeutung benutzt. Dazu zählen Schlüsselwörter und Wörter, welche vom Entwerfer zur Bezeichnung von Entitäten, Beziehungen, Attributen, Werten, Akteuren etc. ausgewählt wurden. Gewöhnlich haben diese Wörter eine bestimmte Bedeutung, und manchmal stimmt ihr Gebrauch in den Anwendungsbereichen mit dieser Bedeutung überein. In einem Beitrag von Buitelaar und van de Riet [8] wird beschrieben, wie ein Lexikon, welches diese Wörter und ihre Bedeutung enthält, zusammen mit einem CASE (Computer Aided Software Engineering) - Tool erfolgreich eingesetzt werden kann.

Ein Werkzeug zum konzeptionellen Entwurf wird in [43] vorgestellt, das Bedarfsspezifikationen in natürlicher Sprache in Konstrukte des erweiterten Entity-Relationship-Approach transformiert. Ein Parsing-Algorithmus in Verbindung mit einer Grammatik und einem Lexikon wird entworfen, um den Anforderungen an das Werkzeug zu genügen. Die Ergebnisse des Parsings werden anhand von Regeln und Heuristiken weiter verarbeitet, womit eine "Beziehung" zwischen linguistischem Wissen und Entwurfsergebnissen aufgebaut wird. Das Werkzeug arbeitet interaktiv, um mehrdeutige, unvollständige und redundante Informationen handhaben zu können.

In seinem Buch "Linguistically Motivated Principles of Knowledge Base Systems" [48] beschreibt Weigand verschiedene Komponenten eines wissensbasierten Informationssystems von einem linguistischen Standpunkt aus. Dabei wird die Theorie der funktionalen Grammatik [14] als ein allgemeiner Rahmen übernommen. In der funktionalen Grammatik werden Strukturen und Eigenschaften der Ausdrücke in Form von Prädikaten repräsentiert. Aus einer wissensbasierten Perspektive sind diese Prädikate informationstragende Behauptungen, und die Prädikatensprache kann als höhere Wissensrepräsentationssprache betrachtet werden. Dies legt eine umfassende Integration von linguistischer und wissensbasierter Forschung nahe.

Im Projekt LILOG (linguistic and logic methods and tools) geht Gerstl [17] davon aus, daß "natürlichsprachliche Semantik" in Systemen zum Textverstehen abhängig ist von der linguistischen Oberflächenstruktur von Sätzen und dem Wissen über den Anwendungsbereich. Während linguistische Theorien dazu tendieren, den erstgenannten Aspekt zu untersuchen, scheinen Entwicklungen in der KI (Künstliche Intelligenz) den letztgenannten zu favorisieren. Die Arbeit in der Knowledge-Engineering-Group des LILOG-Projekts [21] an einem System zum Textverstehen, welches beide Aspekte integriert, gibt den Anlaß zu seinem Beitrag [17]. Das Hauptziel darin ist es, einige Schwierigkeiten bei der Verbindung von Anwendungswissen und Wissen über die Sprache aufzuzeigen und einige Möglichkeiten ihrer Überwindung vorzuschlagen.

Saeki et al. [39] stellen einen Prozeß zur inkrementellen Herleitung einer formalen Spezifikation, ausgehend von einer informalen Spezifikation in natürlicher Sprache, vor. Durch "Entwurfsaktivitäten" wird die Struktur der Softwaremodule, basierend auf einem objektorientierten Modell, von informalen englischen Beschreibungen extrahiert. Jedes Wort in den natürlichsprachlichen Sätzen wird mit einem Softwarekonzept, z. B. Klasse, Attribut oder Methode, verbunden. Besonderes Augenmerk wird Typen von Verbmustern geschenkt, die in Sätzen auftreten, um das System in seinem dynamischen Verhalten zu bestimmen. In "Ausführungsaktivitäten" wird der abgeleitete Modulentwurf verfeinert und präzisiert. Durch den "Entwurfs-/Ausführungs-Aktivitätenzyklus" kann die formale Spezifikation vervollständigt werden.

Die Fähigkeit von Computer-Anwendungen, Informationen zu verarbeiten und mit Benutzern zu kommunizieren sind stark eingeschränkt durch ihre

Unfähigkeit, "freiformatig" geschriebene oder gesprochene Sprache zu analysieren oder zu erzeugen. Während die Techniken, welche sich mit sehr beschränkten "formalen" Sprachen befassen, gut verstanden werden, sind Techniken, welche sich mit geschriebenem Text oder gesprochenem Wort befassen, noch nicht ausgereift. Die Verarbeitung solcher Informationen wird gemeinhin als "Natural Language Processing" (NLP) bezeichnet. Das LOLITA-System (large-scale object-based linguistic interactor and analyser) ist ein prototypisches NLP-System, das geschriebenes Englisch analysieren und erzeugen kann [32]. Einerseits dient LOLITA beim Informationssystementwurf der Vorverarbeitung, um Text zu korrigieren, zu selektieren und zu normalisieren. Andererseits unterstützt LOLITA die "automatische" Konstruktion objektorientierter Modelle anhand der natürlichsprachlichen Bedarfsspezifikation, repräsentiert in Form semantischer Netze [26].

In [52] wird ein Rahmen für ein System vorgestellt, welches ein Programm in einer natürlichen Sprache spezifiziert. Mit dem Ansatz wird eine neue Modularisierungsmethode, basierend auf einer lexikalischen Zerlegung, vorgestellt. Wörter in einer natürlichen Sprache korrespondieren mit den meisten "programmiersprachlichen" Notationseinheiten. Die Semantik der Wörter wird zunächst durch Sätze in natürlicher Sprache festgelegt, welche in Ausdrücke einer Modallogik übersetzt werden, so daß eine exakte Semantik bestimmt und eine semantische Untersuchung am Computer möglich wird. Der Übersetzungsmechanismus basiert auf dem Konzept der Montague-Grammatik [28]. Mehrere Beispiele zur Übersetzung für funktionale und dynamische Systemspezifikationen werden gezeigt. Auch wird eine Software-Entwicklungs-Umgebung (SEU) für das vorgestellte System präsentiert.

Ein Ansatz von Rolland und Proix [38] basiert auf der Prämisse, daß Wissensakquisition stark mit der Manipulation von Sprache zusammenhängt. Er stellt einen Versuch dar, die Sammlung von Aussagen zur Spezifikation einer Problemstellung, ihre Interpretation und ihre Modellierung durch den Gebrauch einer linguistischen Methode zu verbessern. Es wird vorgeschlagen, die Problemspezifikation für die Informationssystementwicklung durch Sätze in natürlicher Sprache auszudrücken. Es wird gezeigt, wie mit einem linguistischen Ansatz, der auf dem CASE-Gedanken beruht, automatisch die Modellierung eines Informationssystems durchgeführt werden kann. Der linguistische Ansatz und seine Implementierung als Expertensystem - bekannt als OICSI (französisches Akronym für "intelligentes Werkzeug für die Informationssy-

stementwicklung") - werden im Detail erläutert. Die zentrale These lautet, daß der linguistische Ansatz allgemeingültig ist, und zwar in zweifacher Hinsicht. So kann er an verschiedene Modellierungstechniken angepaßt und zusätzlich für ein weites Problemspektrum genutzt werden.

2.2. Empirisch-experimentelle Methoden der Informationssystementwicklung

Empirisch-experimentelle Methoden der Informationssystementwicklung haben zum Ziel, fachliche Entwicklungsergebnisse in der Praxis zu erproben (zu überprüfen) oder sie überhaupt erst aus der Praxis heraus schrittweise zu erarbeiten. In diesem Sinne werden konstruktive Entwicklungsmethoden - wie der nächste Abschnitt zeigen wird - auch "experimentell" eingesetzt. Jedoch wird bei ihnen nicht das Ziel verfolgt, eine bestehende Sprache zu analysieren, sondern eine neue, zu Benutzerfachsprachen komplementäre Fachsprache, eine sogenannte "Orthosprache" [29], zur Entwicklung und Nutzung von Informationssystemen aufzubauen.

Ein Fachentwurf kann überprüft werden, indem man in den fachbegrifflich festgelegten Grenzen das Geschehen in einem Anwendungsbereich simuliert. Handeln und Sprechen fallen dabei zusammen, sei es durch Beobachter (Protokollierung) und handelnde Personen getrennt oder von handelnden Personen selbst ausgeführt. Zur Überprüfung relevanter Aussagen beim Systementwurf werden Sprechhandlungen in Verbindung mit physischen Ausführungen (z. B. mit einem Bleistift schreiben und dabei das Wort "schreiben" aussprechen) gemeinsam ausgeführt.

Im anderen Falle (nur Sprechhandlungen liegen vor) wird von Entwicklern beim Systementwurf die Bedeutung relevanter Fachtermini in Gesprächen mit Benutzern **über** Handlungen und ihre Gegenstände in Anwendungsbereichen ermittelt. Hier ist die "naturwüchsige" Kontrolle der Aussagen (Sprechhandlungen) durch das begleitende konkrete Handeln nicht mehr möglich, und es ist nicht ausgeschlossen, daß sich Mißverständnisse ergeben.

In Anlehnung an den Sprachwissenschaftler Bühler [7] wird der erste Ansatz "experimentell-empraktisch" (empraktisch = mit der Praxis) und der zweite Ansatz "experimentell-epipraktisch" (epipraktisch = nach der Praxis) genannt.

Erhebungsverfahren wie die Simulation von Arbeitssituationen in Anwendungsbereichen - mit sprachlich begleitender Protokollierung der Sachverhalte durch ausführende Personen oder Beobachter - eignen sich zur empirischer Analyse eines Fachentwurfs. Dagegen kann z. B. durch Interviews, Fragebögen oder "Beispieldiskussionen" (auf rein sprachlicher Ebene) ein Fachentwurf epipraktisch überprüft werden.

Die vollständige und richtige Angabe der Semantik ist dabei insbesondere für große Anwendungen eine der schwierigsten Aufgaben im Datenbankentwurf. Im Rahmen des Projektes "Rapid Application Database Development" (RADD) werden Erfahrungen bei der informalen Modellierung von Strukturen einer Datenbank gesammelt. Mit dem in [2] vorgestellten Werkzeug wird die Semantik durch eine Analyse natürlichsprachlicher Beschreibungen, die Auswertung von Beispieldaten und Nachfragen in Form einer "Beispieldiskussion" (epipraktisch) gewonnen. Aus natürlichsprachlicher Information können Kandidaten geltender "Constraints" (Integritätsbedingungen) für den Datenbankentwurf abgeleitet werden. Durch Beispiele kann die intendierte Bedeutung der ermittelten Constraints geklärt und gemeinsam mit dem Experten validiert werden. Der Dialog läßt sich in einer einfachen und auch für (formal) ungeübte Experten verständlichen (natürlichsprachlichen) Form bis zur vollständigen Erfassung der Semantik fortführen.

In [27] werden Schwächen in einigen neueren Modellen zur Spezifikation von Büroinformationssystemen diskutiert. Ausgehend von der Überlegung, daß ein Informationssystem insbesondere eine Mittlerrolle in der organisationsinternen Kommunikation spielt, wird ein formales Modell eines Informationssystems aufgebaut. Dazu werden die Sprechakttheorie [5], [41] und neuere Ergebnisse der Organisationstheorie [49] herangezogen. Diese Theorien werden an Informationssysteme dahingehend angepaßt, daß sie eine formale Interpretation derselben liefern. Nicht die technische Spezifikation wird ermittelt, sondern die formalen Strukturen sozialer Interaktion mittels eines Informationssystems werden als Ausgangsbasis für ihren Entwurf erarbeitet.

Die Entwicklung eines (Büro-)Informationssystems geschieht gewöhnlich aus einer ganz bestimmten Perspektive, die die Entwurfsfragen, welche gestellt werden, sowie die Lösungen, welche gesucht werden, von vornherein beeinflußt. In [50] führt Winograd eine Perspektive ein, welche auf der Interpretation der Sprache als Handlung beruht. Die Konsequenzen für den Systement-

wurf werden untersucht. Es wird das Kommunikationstool "The Coordinator" beschrieben, das aus einer Sprache-/Handlungsperspektive heraus entworfen wurde. Dabei wird angedeutet, wie einige Aspekte koordinierter Arbeit mit dem Tool analysiert werden können. Die Sprache-/Handlungsperspektive wird anhand eines Beispiels illustriert, welches auf Studien über die Arbeit von Stationsschwestern im Krankenhaus beruht, und sie wird mit anderen aktuellen, verbreiteten Perspektiven verglichen.

In [4] stellen Auramäki et al. Methoden und Prinzipien dar, die dazu beitragen, Büros als Systeme kommunikativer Handlungen [19] zu untersuchen. Im Rahmen einer kommunikativen Handlung erzeugen Handelnde Bindungen durch symbolische Mittel. Der SAMPO-Ansatz (speech-act based office modeling approach), welcher die Büroaktivitäten als eine Folge von Sprechakten untersucht, die (symbolisch repräsentierte) Bindungen erzeugen, aufrecht erhalten, verändern, offenlegen und beenden, wird vorgestellt. Die wesentlichen Stufen und Methoden bei der Spezifikation eines Büroinformationssystems werden herausgehoben und ihre Anwendung anhand eines einfachen Beispiels illustriert. Abschließend werden Vor- und Nachteile des Ansatzes genannt und einige Schwerpunkte zukünftiger Forschung empfohlen.

Nach Andersen [3] stellen semiotische Ansätze die eigene Interpretation der Benutzer bezüglich ihrer Arbeit in den Vordergrund und nicht diejenigen von Methoden und Werkzeugen, welche vorgeben, eine objektive Beschreibung des Arbeitsprozesses zu bieten. Methode, Ergebnisse und Grenzen eines dieser Ansätze werden untersucht. Eine Schlüsselannahme des vorgeschlagenen Ansatzes besteht darin, daß Informationssysteme sinnvollerweise als Medium der sozialen Interaktion und nicht als Modelle der Realität betrachtet werden sollten. Anhand des Studiums der Sprache, und wie sie in Bezug zu Arbeitssituationen steht, lassen sich dabei Entwurfsideen ableiten, die aus der Wahrnehmung sich wiederholender Muster, Konzepte und Metaphern in der Art, wie Benutzer über ihre Arbeit sprechen, resultieren. Die Ergebnisse dieser Analyse vertiefen das Verständnis der Entwerfer von dem Anwendungsbereich, für den sie entwerfen.

2.3. Konstruktive Methoden der Informationssystementwicklung

Konstruktive Methoden der Informationssystementwicklung haben den Aufbau einer - zu der vorhandenen Fachsprache eines Anwendungsgebiets kom-

plementären - neuen quasi-natürlichen, reglementierten Fachsprache zum Ziel. Sie ist mit Beteiligung der Informationssystem-Benutzer so zu rekonstruieren, daß sich mit ihr alle entwicklungsrelevanten Sachverhalte eines Anwendungsgebiets klar, eindeutig und wörtlich (mit einer normierten Terminologie) darstellen lassen. Daher basieren konstruktive Entwicklungsmethoden i. d. R. auf einem materialen Sprachansatz. Die rekonstruierte und zur Entwicklung von Informationssystemen eingesetzte (materiale) Sprache besteht aus einer Grammatik (Satzbildungsregeln) und einem aus normierten Fachwörtern aufgebauten Fachlexikon (normierte Terminologie).

Eine komplette Methodologie zur Entwicklung von Datenbank-Anwendungen nach diesem, in Anlehnung an die konstruktive Wissenschaftstheorie [30] als "sprachkritisch" bezeichneten Ansatz wurde zuerst in [46] vorgestellt. Im Zentrum der Methodologie steht ein auf drei Abstraktionsebenen (fachliche Ebene, konstruktive Ebene, Schema-Ebene) organisierter Entwicklungsprozeß. Ausgangspunkt der Entwicklung ist ein Mangel in der Informationsversorgung eines Anwendungsbereichs, der auf einer fachlichen Ebene zu einer Aufgabenstellung führt. Durch Aufgabenabstraktion gelangt man zu einer konstruktiven Ebene, auf der die tragenden Fachbegriffe der Anwender wegen der Automatisierungsabsicht präzise "orthosprachlich" (ins rechte Lot gebracht) als Objekttypen festgelegt werden. Ein weiterer Abstraktionsschritt führt zur Schemaebene mit strengen Formalerfordernissen für Relationen (Tabellen) und Operatoren (Datenmanipulation). Aus dem neutralen Konzeptionellen Schema (Basistabellen) können Externe Schemata (Sichten auf Basistabellen) anwendungsbezogen gewonnen werden. Sie sind die Grundlage für den Programmentwurf und die Rückkehr zur konstruktiven Ebene. Ein weiterer Konkretionsschritt führt zur Lösung der Aufgabe auf der fachlichen Ebene, womit der methodologische "Kreis" geschlossen wird.

In [36] stellen die Autoren ein in einem Software-Unternehmen auf obigem Ansatz [46], [35] basierendes implementiertes Verfahren zur Datenmodellierung (Objekttypenmethode) vor, bei dem der methodische Entwicklungsprozeß mit der Sammlung relevanter (natürlichsprachlicher) Aussagen über Sachzusammenhänge in den Anwendungsbereichen beginnt. Es folgt die Klärung und Rekonstruktion wichtiger Fachbegriffe, wobei Regeln zur Behandlung von Synonymen, Homonymen, Äquipollenzen, Vagheit und falscher Bezeichner (Begriffswörter) angegeben werden. Aus der sich auf diese Weise ergebenden Aussagensammlung können durch Analyse der Satzstrukturen und unter Be-

rücksichtigung der rekonstruierten Fachbegriffe - z. T. werkzeugunterstützt - zunächst das Grobdatenmodell (Objekttypen und Beziehungen) und anschließend die Ergebnisse der Feindatenmodellierung (Attribute, funktionale Abhängigkeiten, Integritätsbedingungen) abgeleitet werden. Sämtliche Resultate der Entwicklungsarbeit - Aussagen, Objekttypen und Beziehungen, Attribute, Integritätsbedingungen - werden mit einem "Data Dictionary" (Repository, Metainformationssystem oder Entwicklungsdatenbank), das auch Beziehungen zu Ergebnissen späterer Entwicklungsphasen (logischer und physischer Entwurf) herstellt, strukturiert verwaltet. Erfahrungen mit dem Ansatz in dem Software-Unternehmen werden anhand von Kennzahlen dokumentiert.

Für die Objekttypenmethode [36] zur Datenmodellierung wird in [44] ein automatischer Übersetzer (OTMAR: Object-Type Model Translator) zur Übertragung reglementierter natürlichsprachlicher Aussagen in graphische Repräsentationen der Objekttypenmethode vorgestellt. OTMAR wurde mit den UNIX-Standardwerkzeugen LEX und YACC [20] realisiert. Dabei wird, gewissermaßen als "Filter", um kontrolliert von natürlichsprachlichen Aussagen zu syntaktisch (Aussageformen, Satzbaupläne) und semantisch (Fachtermini) normierten Aussagen überzugehen, eine am Objekttypenansatz orientierte spezifische "Orthosprache" als Referenzsprache definiert. Erst die in der "Orthosprache" reglementierten "natürlichsprachlichen" Aussagen werden mit OTMAR in graphische Repräsentationen der Objekttypenmethode automatisch überführt. Mit dem Werkzeug sind Transformationsregeln sowie Konsistenzbedingungen für die Übersetzung der Aussagen von einer Quellsprache (Orthosprache) in eine Zielsprache (Diagrammsprache) implementiert.

Die Entwicklung von Anwendungssystemen erfordert eine systematische Vorgehensweise bereits in den frühen Projektierungsphasen. Eine grundlegende Schwäche der meisten Modellierungsmethoden für den Fachentwurf ist die mangelnde benutzernahe systematische Konstruktion der Entwicklungsergebnisse aus dem Fachwissen der Anwendungsbereiche heraus. In [40] wird ein Ansatz vorgestellt, der die objektorientierte Spezifikation eines Fachkonzepts auf der Grundlage einer Rekonstruktion der Fachbegriffe eines Anwendungsbereichs in Zusammenarbeit mit den Anwendern ermöglicht. Unterteilt in die Aufgabenfelder Spezifikation, Repräsentation und Integration werden Lösungskonzepte für diesen "terminologie-basierten Ansatz" entwickelt und in

ein Vorgehensmodell für den Fachentwurf mit den Phasen "Rekonstruktion" und "Spezifikation" integriert.

Schließlich wird in [37] die Entwicklung einer am Orthosprachenkonzept der konstruktiven Wissenschaftstheorie [29] orientierten methodenneutralen Konstruktionssprache für Informationssysteme (und Organisationsstrukturen) vorgeschlagen. Um sie von formalsprachlichen Ansätzen der Informationssystementwicklung zu unterscheiden, wird die Konstruktionssprache auch "materiales Entwicklungssystem" (in Abgrenzung zu "formalen Entwicklungssystemen") genannt. Sie besteht aus einer Grammatik, in der die zulässigen Aussageformen (Satzbaupläne) zur Bildung von Aussagen über ein Anwendungsgebiet festgelegt sind, und einem Lexikon, das die normierten Fachwörter aus den Anwendungsbereichen konsistent verwaltet. Grammatik und Lexikon werden in dem Beitrag [37] ansatzweise definiert. An der Ausführung dieses Programms wird in einem Projekt KASPER (Konstanzer Sprachkritik-Programm für das Software-Engineering) gegenwärtig gearbeitet.

Die Aufzählung und Beschreibung der verschiedenen Methoden ist nicht vollständig. Eine Bibliographie zu linguistischen Methoden der Informationssystementwicklung ist in Vorbereitung. Auch die Grenzen zwischen den Arbeiten in Tabelle 1 sind für einige Methoden zu eng und für andere vielleicht zu weit gezogen worden. Ziel der Darstellung war es jedoch, von verschiedenen Grundpositionen ausgehend einige Entwicklungslinien für linguistische Methoden der Informationssystementwicklung anhand vorliegender Arbeiten aufzuzeigen.

3. Workshop zum natürlichsprachlichen Entwurf von Informationssystemen

Ein methodisch-natürlichsprachlicher Beginn der Entwicklung von Informationssystemen wird in der Software-Industrie [36] sowie bei Anwendern in manchen Unternehmen [6], [23] bereits praktiziert. Daneben gibt es - wie Tabelle 1 zeigt - zahlreiche Projekte, die eine Entwicklung von Methoden und Modellen sowie die Bereitstellung von Werkzeugen zur Unterstützung des natürlichsprachlichen Entwurfs von Informationssystemen zum Ziele haben. Die GI-Fachgruppe 2.5.2 EMISA (Entwicklungsmethoden für Informationssysteme)

steme und deren Anwendung) wird einen Pfingstworkshop zu diesem Thema in der Ev. Akademie Tutzing vom 28. - 30. Mai 1996 abhalten.

Mit dem Workshop ist geplant, eine Diskussion innerhalb und zwischen Teildisziplinen wie Requirements Engineering, Information Engineering, Datenbank-Anwendungsentwicklung, Wissensrepräsentation, (Computer-)Linguistik und Wissenschaftstheorie anzuregen. Als Themenkomplexe, die zur Orientierung für Beitragseinreichungen dienen sollen, werden

- Grundlagen methodisch-natürlichsprachlicher Systementwicklung,
- Entwicklung einzelner Methoden und ihre Integration in Vorgehensmodelle,
- Werkzeugentwicklungen und Werkzeugeinsatz,
- Projekte bei Anwendern sowie in Software-Unternehmen

vorgeschlagen. Beiträge im Umfang von 15 DIN A4 Seiten (1,5-zeilig) sind in 4-facher Ausfertigung bis zum 19.11.1995 an den Autor dieses Beitrags zu senden. Die Beiträge werden nach Beurteilung durch ein Programmkomitee in einem Tagungsband publiziert und den Teilnehmern vor Workshop-Beginn vorliegen.

Literatur

- [1] **Abbott, Russel J.:** Program Design by Informal English Descriptions. In: Communications of the ACM, Bd. 26, Nr. 11, November 1983, S. 882-894.
- [2] **Albrecht, M.; Buchholz, E.; Düsterhöft, A.; Thalheim, B.:** Ein Werkzeug zur Gewinnung semantischer Constraints aus natürlichsprachlichen Eingaben und Beispieldaten. In: Datenbanksysteme in Büro, Technik und Wissenschaft, GI-Fachtagung, Dresden, 22.-24. März 1995, hrsg. von Lausen, G., Berlin et al.: Springer-Verlag, 1995, S. 441-460.
- [3] **Andersen, P.B.:** A semiotic approach to construction and assessment of computer systems. In: Information Systems Research: Contemporary Approaches and Emergent Traditions, hrsg. von Nissen, H.-E., Klein, H.K. und Hirschheim, R., Elsevier Publishers B.V. (North-Holland), 1991, S. 465-514.

- [4] **Auramäki, E.; Lehtinen, E.; Lyytinen, K.:** A Speech-Act-Based Office Modeling Approach. In: ACM Transactions on Information Systems, Bd. 6, Nr. 2, April 1988, S. 126-152.
- [5] **Austin, J.L.:** How to do things with words, Harvard University Press, Cambridge Mass. 1962.
- [6] **Bechtel, M. et al.:** Datenmodellierung bei der Bühler AG, Uzwil (Schweiz) - Pensionskasse, Universität Konstanz, Informationswissenschaft, Bericht 3/92, Februar 1992.
- [7] **Bühler, K.:** Sprachtheorie, Die Darstellungsfunktion der Sprache, Verlag Ullstein GmbH, Frankfurt 1978.
- [8] **Buitelaar, P.; van de Riet, R.:** The Use of a Lexicon to Interpret ER Diagrams: a LIKE project. In: Entity-Relationship Approach - ER '92, Proceedings, Karlsruhe 7. - 9. Oktober 1992, hrsg. von Pernul, G. und Tjoa, A.M., Berlin et. al: Springer-Verlag, 1992, S. 162-177.
- [9] **Carnap, R.:** Bedeutung und Notwendigkeit - Eine Studie zur Semantik und modalen Logik, Library of Exact Philosophy Bd. 6, Springer Verlag, Wien/New York 1972.
- [10] **Chen, P.:** English Sentence Structure and Entity-Relationship Diagrams. In: Information Sciences, Bd. 29 (1983), Nr. 1, S. 127-149.
- [11] **Chomsky, N.:** Aspekte der Syntax-Theorie, Suhrkamp Verlag Frankfurt am Main/Akademie Verlag, Berlin 1969.
- [12] **Colombetti, M.; Guida, G.; Somalvico, M.:** NLDA: A Natural Language Reasoning System for the Analysis of Data Base Requirements. In: Methodology and Tools for Data Base Design, hrsg. von Ceri, St., Amsterdam et al.: North-Holland, 2. Auflage, 1985, S. 163-179.
- [13] **Dignum, F.; van de Riet, R.P.:** Knowledge base modelling based on linguistics and founded in logic. In: Data & Knowledge Engineering 7 (1991), S. 1-34.
- [14] **Dik, S.C.:** Theory of Functional Grammar, Dordrecht 1979.
- [15] **Fillmore, C.:** The case for case. In: Universals in linguistic theory, hrsg. von Bach, E. und Harms, R.T., New York et al.: Holt, Rinhart and Winston Inc., 1968, S. 1-90.
- [16] **Flores, Fernando; Graves, G.; Hartfield, B.; Winograd, T.:** Computer Systems and the Design of Organizational Interaction. In: ACM Transaction on Office Information Systems, Bd. 6, Nr. 2., April 1988, S. 153-172.

- [17] **Gerstl, P.:** Linking linguistic and non-linguistic information. In: Data & Knowledge Engineering, Bd. 8, Nr. 3, Juli 1992, S. 205-222.
- [18] **Gunia, H.:** Sprachkritische Entwicklung von Expertensystemen, Dissertation, Technische Fakultät der Universität Erlangen-Nürnberg, 1994.
- [19] **Habermas, J.:** Theorie des kommunikativen Handelns, Bd. 1, Frankfurt/M: Suhrkamp, 1981.
- [20] **Herold, H.:** Unix and seine Werkzeuge, Addison Wesley, Bonn 1992.
- [21] **Herzog, O.; Rollinger, C.-R. (Hsg.):** Text Understanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence, Final Report on the IBM Germany LILOG-Project, Springer-Verlag, Berlin et al. 1991.
- [22] **Holmquist, B.; Andersen, P.B.:** Language, Perspectives and Design. In: Design at Work, hrsg. von Greenbaum, J. und Kyng, M., Hillsdale, NJ, 1991, S. 91-119.
- [23] **Irion, A.M.:** Regelwerk und Qualitätscheckliste zur Bildung von Fachbegriffen bei der Entwicklung und Administration einer normierten Unternehmensfachsprache, Diplomarbeit, Universität Konstanz, Informationswissenschaft, Februar 1995.
- [24] **Kristen, G.:** Object-orientation: the KISS method: from information architecture to information system, Wokingham/England et al.: Addison-Wesley, 1994.
- [25] **Lacity, M. C.; Janson, M.A.:** Understanding Qualitative Data: A Framework of Text Analysis Methods. In: Journal of Management Information Systems, Fall 1994, Bd. 11, Nr. 2, S. 137-155.
- [26] **Lehmann, F. (Ed.):** Semantik Networks in Artificial Intelligence, Pergamon Press Ltd., New York 1992.
- [27] **Lehtinen, E.; Lyytinen, K.:** Action based model of information system. In: Information Systems, Bd. 11 (1986), Nr. 4, S. 299-317.
- [28] **Löbner, S.:** Einführung in die Montague-Grammatik, Kronberg 1976.
- [29] **Lorenzen, P.:** Semantisch normierte Orthosprachen. In: Zum normativen Fundament der Wissenschaft, hrsg. von Kambartel, F. und Mittelstraß, J., Frankfurt/M.: Athenäum, 1973, S. 231-249.
- [30] **Lorenzen, P.:** Lehrbuch der konstruktiven Wissenschaftstheorie, Mannheim et al.: B.I.-Wissenschaftsverlag, 1987.

- [31] **Métais, E.; Meunier, J.-N.; Levreau, G.:** Database Schema Design: A Perspective from Natural Language Techniques to Validation and View Integration. In: Entity-Relationship Approach - ER '93, Proceedings, Arlington 15.-17. Dezember 1993, hrsg. von Elmasri, R.A., Kouramajian, V. und Thalheim, B. Berlin et. al: Springer-Verlag, 1993, S. 190-205.
- [32] **Mich, L.; Garigliano, R.:** Object Oriented Modelling using the Natural Language Processing System LOLITA, in: ISCO 3, International Conference on Information System Concepts: Towards a Consolidation of View, Universität Marburg, 28.-30. März 1995, Marburg, 1995, S. A12-1 - A12-2.
- [33] **Müller-Merbach, H.:** Model Design Based on the Systems Approach. In: Journal of the operational Research Society, 34(1983)8, S. 739-751.
- [34] **Müller-Merbach, H.:** Design of Planning Models Supported by Relational Databases. In: Journal of the Operational Research Society, 42(1991)6, S. 522-523.
- [35] **Ortner, E.:** Aspekte einer Konstruktionsprache für den Datenbankentwurf, Dissertation, Fachbereich 1, Technische Hochschule Darmstadt, Mai 1982, als Buch: Darmstadt: S. Toeche-Mittler, 1983.
- [36] **Ortner, E.; Söllner, B.:** Semantische Datenmodellierung nach der Objekttypenmethode. In: Informatik-Spektrum, Bd. 8 (1989), Nr. 1, S. 31-42.
- [37] **Ortner, E.:** MELCHIOS: Methodenneutrale Konstruktionsprache für Informationssysteme, Universität Konstanz, Informationswissenschaft, Bericht 60-94, Konstanz, September 1994.
- [38] **Rolland, C.; Proix, C.:** A Natural Language Approach to Conceptual Modeling, in: Conceptual Modeling, Databases and CASE: an integrated view of information system development, New York et al.: Wiley, 1992, S. 447-464.
- [39] **Saeki, M.; Horai, H.; Enomoto, H.:** Software Development Process from Natural Language Specification. In: Proceedings of the 11th International Conference on Software Engineering, Pittsburg, Pennsylvania (USA): IEEE Computer Society Press, 1989, S. 64-73.
- [40] **Schienmann, B.:** Fachentwurf mit TAOS: ein Terminologiebasierter Ansatz für die Objektorientierte Spezifikation, Universität Konstanz, Informationswissenschaft, Bericht 67-95, März 1995.
- [41] **Searle, J.R.:** Speech Acts, Cambridge: Cambridge University Press, 1969.

- [42] **Sowa, J. F.:** Conceptual Structures: Information Processing in Mind and Machine, Reading/Massachusetts et al.: Addison-Wesley, 1984.
- [43] **Tjoa, A Min; Berger, L.:** Transformation of Requirement Specifications Expressed in Natural Language into an EER Model. In: Entity-Relationship Approach - ER '93, Proceedings, Arlington 15.-17. Dezember 1993, hrsg. von Elmasri, R.A. Kouramajian, V. und Thalheim, B., Berlin et. al: Springer-Verlag, 1993, S. 206-217.
- [44] **Vogler, M.:** OTMAR - Ein automatischer Übersetzer zur Konstruktion von Objekttypen und Beziehungen aus Aussagen, Diplomarbeit, Universität Konstanz, Informationswissenschaft, November 1994.
- [45] **Wedekind, H.; Ortner, E.:** Systematisches Konstruieren von Datenbankanwendungen - Zur Methodologie der Angewandten Informatik, Carl Hanser Verlag, München 1980.
- [46] **Wedekind, H.:** Datenbanksysteme I: eine konstruktive Einführung in die Datenverarbeitung in Wirtschaft und Verwaltung, 2., völlig neu bearbeitete Auflage, Mannheim et al.: B.I.-Wissenschaftsverlag, 1981.
- [47] **Wedekind, H.:** Objektorientierte Schemaentwicklung: ein kategorialer Ansatz für Datenbanken und Programmierung, Mannheim et al.: BI-Wissenschaftsverlag, 1992 (Reihe Informatik; 85).
- [48] **Weigand, H.:** Linguistically Motivated Principles of Knowledge Base Systems, Dordrecht et al.: Foris Publications, 1990.
- [49] **Williamson, O.E.:** Transaction-cost economics: the governance of contractual relations. In: J. Law Econ., 22(1979)2, S. 233-261.
- [50] **Winograd, T.:** A Language/Action Perspective on the Design of Cooperative Work. In: Human-Computer Interaction, Bd. 3 (1987-1988), S. 3-30.
- [51] **Wintraecken, J.J.V.R.:** The NIAM Information Analysis Method: Theory and Practice, Dordrecht et al.: Kluwer Academic Publishers, 1989.
- [52] **Yonezaki, N.:** Natural Language for Requirement Specification. In: Japanese perspectives in software engineering, hrsg. von Matsumoto und Ohne, Singapore et al.: Addison-Wesley, 1989, S. 41-71.

Modellierung in soziotechnischen Systemen – Von Menschen, Organisationen, Modellierern und Modellen

Dr. Hansjürgen Paul
Wissenschaftszentrum NRW
Institut Arbeit und Technik
Gelsenkirchen

H. Paul

Kaum ein Thema erfährt in Buchveröffentlichungen und Zeitschriftenartikeln in der letzten Zeit eine solche Beachtung wie Workflowmanagementsysteme. Dabei könnte der Eindruck entstehen, daß es eine große Anzahl von Betrieben geben muß, die weder einen Geschäftsprozeß noch einen Workflow zu haben scheinen und dringend eines „business reengineering“ mit anschließender Entwicklung von Workflows und Workflow-Anwendungssystemen bedürfen. Warum bei der Realisierung dieser Lösungen dann mitunter haargenau an dem, was die Menschen als eigentliche Träger von Geschäftsprozeß und Workflow benötigen, vorbeigearbeitet und den Bedürfnissen der Organisationen nicht adäquat Rechnung getragen wird, soll in diesem Beitrag näher diskutiert werden. Dabei werden erste Ergebnisse der EMISA-Arbeitsgruppe „Modellierung in soziotechnischen Systemen“, deren Gründungstreffen am 20. und 21. April 1995 in Gelsenkirchen stattfand, dargestellt. Diese Resultate werden durch Erkenntnisse, Sichtweisen und Positionen der Abteilung Produktionssysteme im Institut Arbeit und Technik ergänzt.

Die Herausforderung

Der Stellenwert, der Begriffen wie Geschäftsprozeßmodellierung, Workflowmanagementsystemen oder „business reengineering“ nicht nur in der Fachpresse eingeräumt wird, könnte den Eindruck erwecken, dieses Thema sei ein neues wissenschaftliches Themengebiet, das erst durch den Boom der Computerbranche praxisrelevant geworden ist und einen ähnlichen Markt verspricht, wie vor Jahren PPS-Systeme. Tatsache ist jedoch, daß Wirtschaftsunternehmen bereits wesentlich länger existieren als Workflowmanagementsysteme im speziellen oder Computersysteme im allgemeinen. Zwar kann man die ersten Computeranwendungen in Versicherungsunternehmen nach dem Zweiten Weltkrieg durchaus als Vorläufer von Geschäftsprozeßsystemen interpretieren, dennoch sind Geschäftsprozesse keine Erfindung des Computerzeitalters. Auch lassen sich Überlegungen zur objektorientierten Organisationsgestaltung bis in die zwanziger Jahre unseres Jahrhunderts zurückverfolgen.

Gemeinsames Motiv für die Diskussionen um Geschäftsprozesse und Arbeitsflüsse ist eher der status quo, den viele Betriebe inzwischen erreicht haben. Angestachelt durch die Erfolge unter den Bedingungen der Massenproduktion haben sie ihre tayloristisch-fordistische Arbeitsorganisation vorangetrieben und u. a. durch Fließbandarbeit, Automation und Robotereinsatz untermauert. Mit rückläufigen Produktionsmengen, immer kleiner werdenden Stückzahlen bis hin zur Kleinstserienfertigung und mit ständig wachsenden Kundenansprüchen an Flexibilität, Qualität, Individualität und Preiswürdigkeit erkennen zahlreiche Betriebe, daß einschneidende Veränderungen – vor allem im organisatorischen Bereich – notwendig sind, um am Markt noch bestehen zu können (vgl. z. B. Brödner / Pekruhl, 1991). Eines der größten Probleme ist dabei nur zu oft das Korsett, das sich solche Betriebe durch die EDV-Systeme angelegt haben; die Kürzel CIM, PPS und BDE sollen hier mit ihren bekannt-unerwarteten Risiken und Nebenwirkungen für sich stehen.

Nicht erst bei der Reorganisation des Gesamtunternehmens, sondern bereits bei kleinen Veränderungen der Betriebsstruktur zeigt sich, daß aus den einstmals stützenden Korsetts inzwischen Stolperfallen und Fesseln geworden sind; selbst unscheinbare, kleine Verände-

rungen schaffen unerwartete Probleme. Dies gilt umso mehr, wenn es um organisatorische Veränderungen geht, die die Entscheidungsstrukturen verändern und beispielsweise dezentrale Konzepte in einem streng hierarchisch ausgerichteten Betrieb Einzug halten.

Diese Situation ist die eigentliche Ausgangssituation der Geschäftsprozeßmodellierung und des Workflowmanagements, die eigentliche Herausforderung für diese Disziplin der Informatik: nicht *nach* erfolgter Reorganisation für den neuen Betrieb ein neues Korsett zu zimmern, sondern mitzuhelfen, die angeschlagene Organisation von ihren Fesseln zu befreien und ihr ein wirksames Instrumentarium an die Hand zu geben, um sich schnell und flexibel auf neue Marktanforderungen einstellen und sich selbst den neuen Gegebenheiten anpassen zu können.

Modellierung für Menschen

Nur wenige Ansätze stellen sich der Herausforderung einer menschenzentrierten Modellierung. Oft werden theoretische Konzepte formuliert, die den Eindruck erwecken, daß es um die Konstruktion eines rein technischen Systems geht, vergleichbar einem Getriebe oder einem Motor. Daß es aber darum geht, ein soziotechnisches System zu gestalten, zu entwickeln, zu lenken, wird dabei dann übersehen – oder es wird bewußt von diesen Aspekten abstrahiert. Soziotechnische Systeme – Systeme, in denen eine Menge in Beziehung stehender Menschen und technischer Systeme unter bestimmten Bedingungen nach veränderbaren, aushandelbaren Regeln bestimmte Aufgaben erfüllen – erfordern soziotechnische Methoden und soziotechnische Lösungen.

Die Arbeitsgruppe „Modellierung in soziotechnischen Systemen“ der EMISA hat es sich zum Ziel gesetzt, dem Trend zum Abstrahieren vom einzelnen Arbeitszusammenhang, von den Arbeitsaufgaben und damit dem Abstrahieren vom Menschen entgegenzuwirken und die Rolle des Menschen in den Mittelpunkt der Diskussion um Datenmodellierung, Workflowmanagementsysteme und Geschäftsprozesse zu rücken.

Theorie und Praxis sollen von dieser Arbeitsgruppe gleichermaßen profitieren: die Praxis soll nicht länger auf Antworten zu Fragen warten, die nie formuliert wurden und die Theorie soll nicht länger Antworten erarbeiten, die niemand erfragt hat. Sicherlich gibt es nicht *die* Theorie und *die* Praxis und sicherlich ist es mitunter sehr fruchtbar, Antworten auf Fragen zu entwickeln, die bisher noch niemand gestellt hat. Das Zusammenspiel von modellierender Theorie und modellierender Praxis ist aber längst nicht so gut, daß man sich beruhigt zurücklehnen könnte. Es sei bei dieser Gelegenheit daran erinnert, daß man in einem sehr eng verwandten Terrain inzwischen bereits das silberne Jubiläum einer Krise feiern konnte – und dies trotz intensivster „theoretischer“ Bemühungen.

In einer ersten Runde hat die Arbeitsgruppe „Modellierung in soziotechnischen Systemen“ die Rolle des Menschen in der Modellierung aus drei verschiedenen Perspektiven untersucht: der Mensch als Modellierungsobjekt¹, der Mensch als Nutzer von Methoden und Werkzeugen und der Mensch als Kunde bzw. Benutzer der zu entwickelnden Produkte. Im Fall des Modellierungsobjekts wird der Mensch als zu verplanende Größe angesehen, als Ressource, als Funktionsträger, als Ausführer bestimmter Arbeitsschritte. Arbeitsschritte, Arbeitsaufgaben, Arbeitszusammenhänge werden für ihn entworfen, verknüpft und verplant. Datenmodellierung zählt hierzu, aber auch das Workflowmanagement. Der Nutzer von Methoden und Werkzeugen wendet die unterschiedlichen Verfahren an, um beispielsweise Dialoganwendungen für interaktive Systeme zu entwickeln, aber auch um Unternehmensmodellierung zu betreiben; hierbei tritt der Mensch als aktiver Gestalter, als Entwickler, als Realisierer in Erscheinung. In der dritten Betrachtungsebene schließlich

¹ Auch wenn es für manchen nicht so klingen mag: *Modellierungsobjekt* ist wertfrei und nicht diskriminierend gemeint. Der Mensch (Arbeitnehmer, Mitarbeiter, Angestellter, Benutzer, ...) wird in dieser Perspektive zur abstrakten Größe; Konzepte wie Mitbestimmung, Partizipation oder Kooperation werden dadurch keineswegs ausgeschlossen.

benutzt der Mensch Systeme, die für ihn entwickelt worden sind, z.B. bei interaktiven Computeranwendungen, oder agiert in Organisationseinheiten innerhalb vorgegebener Schemata.

Zielbereiche sind für diese drei Betrachtungsebenen die Datenmodellierung, die Dialoganwendung, die Unternehmensmodellierung und das Workflowmanagement. Darüber hinaus wurde in der Arbeitsgruppe das Konzept einer Referenzsprache bei der Semantischen Modellierung in Raum und Zeit als Voraussetzung für Sprachvereinheitlichung diskutiert. Natürlich konnte dieses weite Spektrum in der bisherigen Arbeit nicht vollständig und erschöpfend bearbeitet werden. Das beschriebene Raster dient als Orientierungsraster, um die Interessensschwerpunkte in der Arbeitsgruppe zu konkretisieren und die bisherigen Beiträge zueinander in Beziehung zu setzen.

Aus den Diskussionen zu den Themenschwerpunkten hat die Arbeitsgruppe zwei Kataloge – einen Antworten- und einen Fragenkatalog – erarbeitet, die den Konsens der Gruppe zusammenfassen und als Orientierung für die weitere Arbeit der Gruppe dienen sollen. Die gesammelten Antworten beziehen sich dabei nicht etwa auf die formulierten Fragen, vielmehr leiten sich viele der Fragen aus den Antworten ab. Auch auf die Gefahr hin, daß Ungleichgewichte entstehen oder die gefundenen Antworten durch das Herauslösen aus dem Diskussionszusammenhang banal erscheinen mögen, sollen einige hier wiedergegeben werden. Die vollständigen Kataloge sind Bestandteil des Sitzungsprotokolls, das über den Autor dieses Beitrags bezogen werden kann.

Bezüglich des Modellierungsobjekts wurde festgestellt, daß Arbeitshandeln modelliert werden muß, wobei die Zielsetzung der Modellierung von Arbeitshandeln sich signifikant von der Zielsetzung bei der Entwicklung von Software unterscheidet und nicht mit ihr gleichgesetzt werden darf. Daraus leitet sich die Konsequenz ab, daß die Methoden der Modellierung nicht die der Software-Entwicklung sein können. Eine weitere gemeinsame Antwort betrifft die Modellierung unmittelbar. Demnach heißt Modellierung, eine Verbindung von sozialem Gebilde und technischem System zu schaffen. Dabei ist der Mensch Teil eines Systems mit Freiheitsgraden bei der Erfüllung von Anforderungen. Die Handlungsregulationstheorie der Arbeitspsychologie mit ihren Aussagen über das Handeln von Individuen und Gruppen wurde als wichtiges Hilfsmittel bei der Modellierung in soziotechnischen Systemen erkannt.

Für die Nutzung der Methoden wurden Anforderungen an die Modellierung in soziotechnischen Systemen formuliert: Unvollständigkeit, die Existenz vermittelnde Zwischenebenen und entsprechender Operatoren, die Integration von Daten- und anderen Aspekten. Dabei wird Modellierung in soziotechnischen Systemen als ein kreativer und stetig detaillierender Prozeß verstanden, der auch den „Mut zur Lücke“ für notwendige Freiheitsgrade fordert.

Die in der Arbeitsgruppe erarbeiteten Fragen berühren zum großen Teil Themengebiete, in denen der Praktiker vermeintlich konkrete Antworten der Modellierungstheorie erhofft. So z.B. die Frage nach den Grenzen der Modellierung: was *kann* ich modellieren, was *muß* ich modellieren? Inwiefern kann man die Kooperation von Menschen überhaupt modellieren? Wieweit sollte bzw. muß man dabei überhaupt gehen? Grundsätzlicher: was müssen Modellierer leisten? Was gehört alles zu ihren Aufgaben, z.B. die Sprachvereinheitlichung der Beteiligten?

Ähnliches gilt für die Frage nach „verwertbaren“ wissenschaftlichen Konzepten. Welche Konzepte der Arbeitswissenschaft sind für die Prozeßmodellierung hilfreich? Welche Methoden, welche Werkzeuge helfen bei der Modellierung des Wandels von Organisationen? Wie kann man die Organisation, wie die Administration, wie die Kreation des Wandels in soziotechnischen Systemen unterstützen? Durch Workflowmanagementsysteme? Oder benötigt man mehr?

Aus solchen und ähnlichen Fragen ergeben sich dann auch Fragen, die die unmittelbaren Arbeitsgegenstände des Modellierers betreffen. Wie sehen z.B. Werkzeuge des organisatorischen Handelns, d.h. des Handelns von Organisationen in Unterscheidung zu Individuen und Gruppen, aus? Aber auch die eigene Sichtweise wird reflektiert. Welche Bedeutung hat

das Menschenbild, z.B. ein mechanistisches Menschenbild, für die Qualität eines Modells, z.B. für die Wirtschaftlichkeit? Worin unterscheidet sich eigentlich *im Modell* eine Maschine von einem Sachbearbeiter?

Darüber hinaus hat die Arbeitsgruppe eine Reihe von „Metafragen“ und „Metaantworten“ formuliert, die übergeordnet thematischen Treibstoff für die Arbeitsgruppe beinhalten. Dabei sind „Fangfragen“ wie die nach der Sicherung von Flexibilität ebenso vertreten, wie die offenen Fragen nach den Bewertungskriterien für Methoden und ihre Tauglichkeit bzw. Angemessenheit oder den Konsequenzen der Erkenntnisse über die Modellierung in soziotechnischen Systemen für die Software von morgen. Informationssysteme sind adäquat zur Unternehmensrealität zu modellieren und müssen daher flexibel anpaßbar sein. Methodisch wäre *ein* Rahmen für die Orientierung bei der Modellierung von soziotechnischen Systemen wünschenswert. Ob dieser existieren kann, ob er dann noch handhabbar bzw. hinreichend aussagekräftig ist, gehört zu den offenen Fragen, denen sich die Arbeitsgruppe in ihrer weiteren Tätigkeit stellen will.

Handeln durch Workflows

Ausgangspunkt der Überlegungen der Arbeitsgruppe „Modellierung in soziotechnischen Systemen“ war die Einschätzung, daß es in der modellierenden Praxis um weit mehr geht, als um die Konstruktion deterministischer, rein technischer Systeme. Unabhängig davon, ob es um Unternehmensmodellierung, Workflowmanagement, Datenmodellierung oder Anwendungsentwicklung geht, stets gilt es, das Zusammenwirken von Mensch *und* Technik zu entwickeln, zu verändern, zu verbessern. Es wird in soziale Prozesse eingegriffen, Kommunikationsstrukturen werden zerstört, verändert oder neu geschaffen, Arbeitszusammenhänge aufgelöst, manipuliert oder definiert. Dies bedeutet, daß die Verantwortung des Modellierers für den Gesamtprozeß und die realisierte Lösung über das Verständnis einer reinen Produkthaftung hinausgeht (vgl. dazu die ethischen Leitlinien der GI).

Modellieren meint das Gestalten von Arbeitshandeln, d.h. es werden Handlungsziele entwickelt, Arbeitsmittel und Arbeitsgegenstände vorgesehen, Arbeitsergebnisse vorgegeben. Über die Unterstützung von Arbeitshandeln durch interaktive Computersysteme liegen der Informatik durch die Arbeiten der letzten fünfzehn Jahre gesicherte Erkenntnisse vor. So ist die Disziplin der Software-Ergonomie aus dem Bemühen heraus entstanden, menschengerechtere und den Arbeitsaufgaben entsprechende Systeme zu entwickeln. Man weiß, welche Eigenschaften Computersysteme für system- bzw. benutzergeführte Interaktion aufweisen sollten und wie diese zu gestalten sind. Dabei geht es jedoch im wesentlichen um die Benutzung *eines* interaktiven Systems durch *einen* Benutzer. Zwar sind die Computersysteme vernetzt und der Benutzer teilt sich bestimmte Ressourcen mit anderen Benutzern – es geht jedoch fast ausnahmslos um die Unterstützung individuellen Arbeitshandelns.

Sobald es aber gilt, mehr als den einzelnen zu unterstützen, beispielsweise Gruppenarbeit durch Groupware zu unterstützen, gehen die Meinungen über die Gestaltung dieser Systeme stark auseinander. Zur Zeit diskutiert man, welche der bekannten Kriterien von konventionellen Systemen auf Groupware übertragen werden können bzw. welche zusätzlich zu erarbeiten sind, z.B. „Gruppengerechtigkeit“, oder welche Gestaltungsforderung eventuell für Groupware keine Gültigkeit besitzen (vgl. Oberquelle, 1993; Böcker, 1995).

Softwaresysteme für Geschäftsprozesse bzw. Arbeitsflüsse gehen noch einen Schritt weiter. Sie sind nicht nur Systeme für den einzelnen Benutzer oder eine kooperierende Gruppe, sie sind Werkzeuge² für Organisationen („Organisationware“). Sie sollen das Wirken eines Betriebes oder eines ganzen Unternehmens unterstützen, sollen Hilfsmittel für organisa-

² Software muß nicht zwangsläufig „Werkzeug“ sein, alternative Metaphern, wie etwa die Vorstellung vom interaktiven System als vermittelndes Medium zwischen Menschen, sind sicherlich interessante Sichtweisen – insbesondere für Groupwaresysteme (vgl. dazu Nake, 1986; Schelhowe, 1993; Bolz / Kittler / Tholen, 1994).

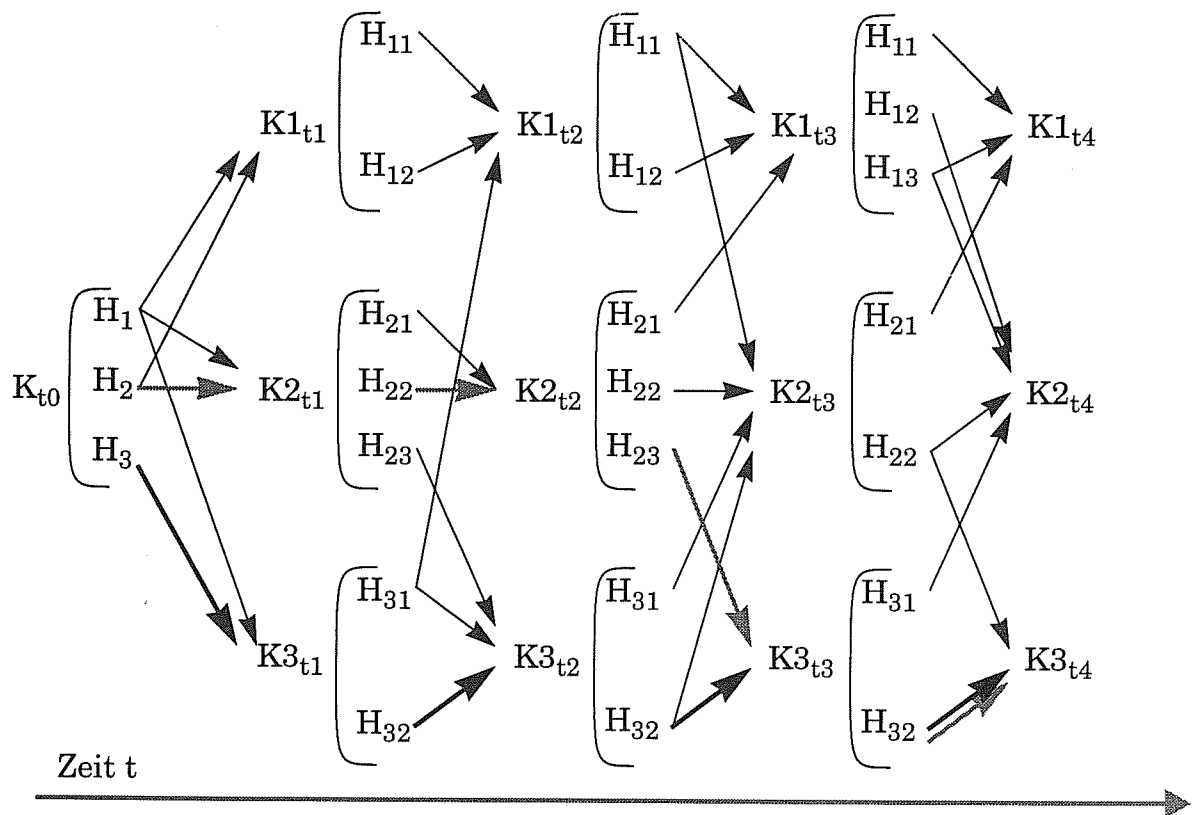


Abb. 1: Graphische Darstellung eines Handlungsfeldes mit Handlungsalternativen H und ihren Konsequenzen K (nach Oesterreich, 1981: 100 und Oesterreich, 1982)

tionales Handeln sein. Die Erkenntnis, daß Organisationen *handeln*, ist dabei längst nicht selbstverständlich, sah man in ihnen doch lange Zeit eher statische Gebilde (vgl. Ortmann et al., 1990; Sydow, 1992; Küpper / Ortmann, 1992; Ortmann, 1995). Wie nun ein solches System aussehen könnte, welche Funktionen es seinem Benutzer zur Verfügung stellen sollte, dies ist noch mehr „Neuland“ für die Informatik als die Ergonomie von Groupware.

Für den Bereich der Unterstützung individuellen Arbeitshandelns ist es möglich, arbeitspsychologische Erkenntnisse über menschliches Handeln – und damit Kenntnisse über die optimale Unterstützung dieser Handlungen – auf die ergonomische Gestaltung interaktiver Systeme zu übertragen. Dies hat seinen Niederschlag beispielsweise in den Grundsätzen der Gestaltung explorationsfreundlicher Systeme gefunden (vgl. Paul, 1995). Daß eine Übertragung bestimmter Erkenntnisse über menschliches Handeln vom Individuellen zum Gemeinschaftlich-Kooperativen möglich ist, erscheint wahrscheinlich, so etwa das Konzept der kollektiven Handlungsregulation (vgl. Oesterreich, 1981: 167 ff.).

Viele Darstellungen von Geschäftsprozessen (z. B. in Wiborny, 1991; Sinz, 1994; Jablonski 1995; Versteegen, 1995) ähneln in ihrer Struktur den Darstellungen menschlichen Handelns wie sie aus der Arbeitspsychologie bekannt sind, insbesondere ähneln sie jenen der Handlungsregulationstheorie (Oesterreich, 1981; Oesterreich, 1982; vgl. Abb. 1). Es liegt nahe, dieser Ähnlichkeit nachzugehen und beispielsweise zu überprüfen, inwieweit Erkenntnisse über individuelle Handlungsregulation, beispielweise das Fünf-Ebenen-Modell der Handlungsregulation (vgl. Abb. 2), auf organisationales Handeln übertragbar sind und inwieweit sich daraus unmittelbare Modellierungs- bzw. Gestaltungsvorgaben ableiten lassen.

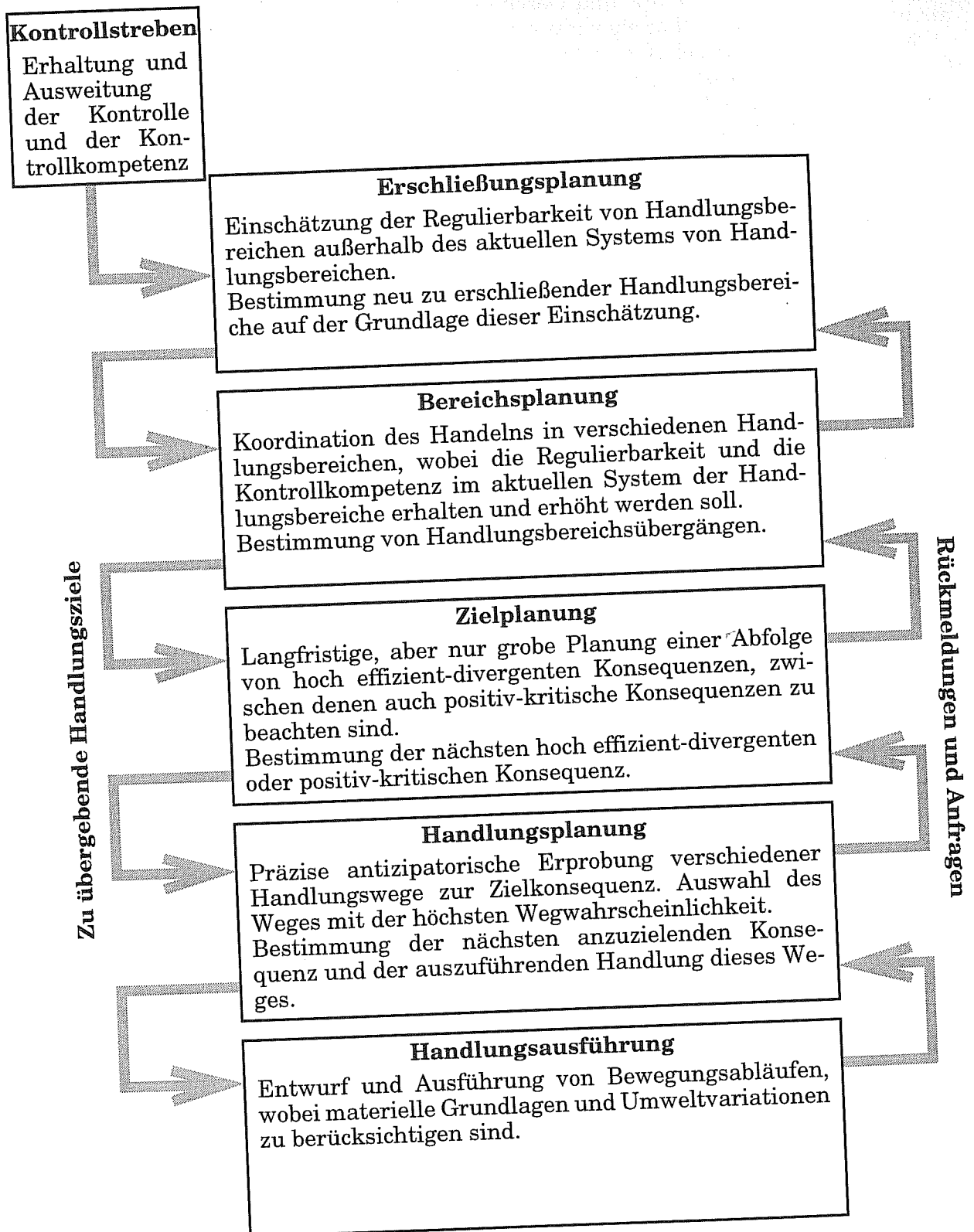


Abb. 2: Das Fünf-Ebenen-Modell der Handlungsregulation (vereinfachte Darstellung nach Oesterreich, 1981: 142f.)

Die Verwandtschaft zwischen Geschäftsprozessen und menschlichem Handeln – verstanden als bewußtes, zielgerichtetes Verhalten – läßt sich noch unmittelbarer aufzeigen. Dazu gilt es, die Begriffe Workflow und Geschäftsprozeß genauer zu fassen und gegeneinander abzugrenzen und das bisher eher intuitive Verständnis dieser Begriffe zu konkretisieren.

Ein Workflow kann als abgrenzbarer, arbeitsteiliger Prozeß zur Erstellung und Verwertung betrieblicher Leistungen angesehen werden. Er stellt einen dynamischen Ablauf mit Initialisierung und definiertem Abschluß dar, der mit dem Verständnis einer Abfolge von Handlungen in der Sichtweise der Handlungsregulationstheorie (vgl. Oesterreich, 1981) verglichen werden kann. Den Workflows übergeordnet sind die Geschäftsprozesse, die die Verfolgung übergeordneter, strategischer Ziele zur Aufgabe haben und den Plänen einer Organisation entsprechend ausgerichtet sind. Workflows sind dabei Mittel zum Erreichen dieser Ziele. Sie entsprechen daher – bezogen auf die Handlungsregulationstheorie – den Zielkonsequenzen von Handlungswegen.

Geschäftsprozesse können als Handlungswege von Organisationen angesehen werden, die dem Erreichen von übergeordneten, strategischen Zielen dienen. Workflows, ihr Betrieb, ihre Definition, ihre Reorganisation, ihre Kombination, sind dann Handlungen von Organisationen zum Erreichen dieser Zielkonsequenzen. Damit ist eine Basis geschaffen, durch die es möglich wird, die Vorstellung von organisationalem Handeln zu konkretisieren. Dies kann natürlich nur eine erste Annäherung sein – keinesfalls soll damit organisationales Handeln dem Betreiben von Geschäftsprozessen gleichgesetzt werden.

Eine Organisation, die in der hier beschriebenen Form handelt, stellt bestimmte Anforderungen an die Hilfsmittel, die sie beim Erreichen der übergeordneten Ziele bzw. Zielkonsequenzen unterstützen sollen. Diese Werkzeuge sollen beispielsweise den Umgang mit Arbeitsgegenständen erleichtern und effizienter machen: sie sollen der Organisation den Umgang mit den Workflows erleichtern, um die Geschäftsprozesse erfolgreich zu betreiben und die Ziele der Organisation zu erreichen helfen. Zusätzlich müssen sie das Unternehmen in seiner Anpassung an den Markt, bei der permanenten Reorganisation aufgrund sich verändernder Strukturen unterstützen. Diese *Workflowmanagementsysteme* müssen es der Organisation ermöglichen, den Workflow wirklich zu managen – nicht nur seinen Fluß zu gewährleisten.

Diese „Organisationware“ muß den Unternehmen Mittel und Wege bereitstellen, sich selbst zu verändern: neue Geschäftsbereiche zu entwickeln und zu realisieren, Geschäftsbereiche zusammenzulegen bzw. aufzulösen, Teilaufgaben zu verlagern, neue Aufgaben zu übernehmen, Entscheidungsverfahren umzuorganisieren, neue Kooperationsformen zu praktizieren. Solche Systeme sind dann nicht länger steife Korsetts, die den Betrieb in eine überkommene Form pressen oder Fesseln, die seine Bewegungsfreiheit einengen. Sie sind vielmehr Instrumentarien, die moderne Organisationen jene Flexibilität zu nutzen erlauben, die sie sich mit der Überwindung tayloristischer Arbeitsorganisationen erarbeitet haben. Betriebliche Reorganisation schafft Potentiale – „Organisationware“ trägt dazu bei, diese Potentiale auch nutzen zu können.

Möglichkeiten

Aus diesen Überlegungen leiten sich eine Reihe von Möglichkeiten ab, die leider nicht sofort und direkt in der modellierenden Praxis Wirkung zeigen. Es sind noch eine ganze Reihe von Zwischenschritten notwendig, um spürbare Resultate zu erbringen.

Wie so oft in der Informatik gilt es zunächst, mehr über den „Benutzer“ und seine „Arbeitsaufgabe“ in Erfahrung zu bringen, also über Organisationen und ihr Handeln. Nur wenn es gelingt, diese Abläufe hinreichend genau zu verstehen, wird es möglich, adäquate Lösungen zu realisieren. Neben industriesoziologischen und arbeitspsychologischen Arbeiten ist hierbei sicher auch die weitere Erforschung von Groupware und CSCW als Bindeglied zwischen dem Handeln von Individuen zum Handeln von Organisationen gefordert.

Erkenntnisse und Lösungen aus diesen Bereichen könnte man beispielsweise iterativ zusammenführen, um so bessere, angemessenere Gesamtlösungen zu erarbeiten.

Nicht zuletzt aus softwaretechnischer Perspektive sind Workflowmanagementsysteme aufgrund ihrer Komplexität eine kaum zu übersehende Herausforderung. Konventionelle, monolithische Systeme sind hierbei keine probaten Mittel. Vielmehr gilt es, die Komplexität beherrschbar und das scheinbar unentwirrbare Netz von Abhängigkeiten und Wirkzusammenhängen transparent zu machen. Hierbei bieten sich Modularisierung und Objektorientierung als adäquate Prinzipien an. Dezentrale Konzepte, die den einzelnen Modulen eine maximale Autonomie zubilligen und die Wahrung von Konsistenz durch standardisierte Schnittstellen und Delegation von Verantwortungen zu erreichen suchen, sind angemessenere Wege – vergleichbar den Methoden und Prinzipien moderner Organisationen.

Die Zementierung von Workflows in vermeintlichen Automatismen, die absolutistisch anmutende Einprogrammierung von betrieblichen Entscheidungsprozessen in Workflow-Anwendungssystemen führt bestenfalls zu einem „Dienst nach Vorschrift“, von dem hinlänglich bekannt ist, daß er praktisch einem Streik gleichkommt. Modellieren in und für soziotechnische Systeme bedeutet, Möglichkeiten zum Handeln zu schaffen. Es kann nicht darum gehen, die Abläufe einer Organisation im Sinne eines konventionellen Algorithmus festzulegen und wie bei prozeduralen Programmiersprachen in einem monolithischen Softwareblock einzufrieren. Vielmehr sollte es das Ziel sein, Verfahren, Methoden, Objekte bereitzustellen, die in Abhängigkeit von der Arbeitsaufgabe in der Organisation flexibel eingesetzt werden können – vergleichbar einer objektorientierten Anwendungsentwicklung. Erst durch diese Modularisierung der Workflow-Anwendungssysteme ist es dann auch möglich, jenen Grad an Flexibilität zu erreichen, der notwendig ist, um die Organisation auch bei ihrer eigenen Umstrukturierung zu unterstützen und erst dadurch wird es möglich, Workflowmanagementsysteme zu schaffen, die ein wirkliches Management der Workflows erlauben.

Workflowmanagementsysteme sollen Hilfsmittel organisationalen Handelns sein. Nachwievor sitzen bei diesen Systemen aber Individuen an den Computern und benutzen sie. Man könnte damit die Vorstellung verknüpfen, daß Workflowmanagementsysteme nichts anderes als „Workflow-Editoren“ sind, mit denen ein Manager an seinem Computer, unterstützt durch eine graphische Benutzungsoberfläche, Workflows auseinandernimmt, löscht, erzeugt, neu zusammenstellt und damit auf Knopfdruck die betriebliche Realität verändert hat. Genau dies dürfen Workflowmanagementsysteme auf keinen Fall sein. Sie sollen den Betrieb in die Lage versetzen, selbständig zu agieren, Workflows zu verändern, zu optimieren, anzupassen – ohne daß dadurch die Gesamtorganisation in Mitleidenschaft gezogen wird, ohne daß dadurch sämtliche Anwendungssysteme wertlos werden. Workflowmanagementsysteme sollen den Betrieben helfen, jene Potentiale zu nutzen, die moderne Organisationskonzepte schaffen.

Daß Workflowmanagementsysteme in dem hier beschriebenen Sinn auch als Hilfsmittel bei dem immer wieder eingeforderten Management des Wandels dienen können, ist längst nicht selbstverständlich. Nimmt man diese Herausforderung an, so wird bald deutlich, daß diese Systeme noch zusätzliche Eigenschaften haben sollten. So sollte es möglich sein, Workflowmanagementsysteme so zu entwickeln, daß ihre Verwendung bestimmte Entwicklungen in Organisationsstrukturen wenn nicht ausschließt, so ihnen doch entgegenwirkt.

Ein Beispiel dafür ist die Trennung von Kopf und Hand. Im Sinne des Fünf-Ebenen-Modells der Handlungsregulation (Abb. 2) bedeutet dies, daß es – aus tayloristischen Traditionen entstanden – eine Tendenz gibt, Handlungsausführung von Erschließungsplanung bzw. Bereichsplanung abzukoppeln, etwa die Ausführung auf die Werkstattebene zu verlagern, während die Entscheidungen in den Managementetagen getroffen werden. Organisationen sollten bei Unternehmensmodellierung und Workflowmanagement so gestaltet werden, daß alle fünf Ebenen der Handlungsregulation, d.h. Erschließungsplanung, Bereichsplanung, Zielplanung, Handlungsplanung und Handlungsausführung, zu jedem einzelnen Workflow und zu jedem Arbeitszusammenhang gehören. Eine solche Organisationsstruktur kann selbstverständlich nicht durch eine wie auch immer geartete Gestaltung von Work-

flowmanagementsystemen garantiert werden. Dennoch sollte es möglich sein, eine Orientierung an einem solchen Leitbild mit Workflowmanagementsystemen zu verknüpfen, zumindest aber dem widersprechende Organisationsentwicklungen zu erschweren.

Nachdem deutlich wurde, was noch nicht möglich ist bzw. was noch zu leisten ist, sei abschließend die Frage gestellt, wie die nächsten Schritte aussehen. Ein wichtiger Punkt ist dabei sicher, organisationales Handeln bewußt zu machen. Es gilt, mehr über organisationales Handeln zu erfahren. Dabei stehen aber – aus der Perspektive der Modellierung – weniger die Strukturen, Mechanismen und Einflußgrößen im Mittelpunkt des Interesses. Wesentlich sind die Handlungen, die es zu unterstützen gilt. Längst nicht alle Handlungen treten gleich häufig auf, was dennoch aber kein Indikator dafür ist, ob eine Handlung wesentlich ist oder der Unterstützung bedarf. So ist die Anpassung an neue Marktanforderungen oder veränderte Zulieferstrukturen sicherlich keine ständig wiederkehrende Aktivität. Nichtdestoweniger sind solche Veränderungen äußerst kritische Vorgänge, deren Scheitern fatale Konsequenzen für eine Organisation haben kann.

Eine hinreichend modulare, flexible Organisation mit adäquaten Workflows zur Abwicklung der Geschäftsprozesse und geeignete Workflowmanagementsysteme, die die Handlungsfähigkeit der Organisation erhöht und nicht zusätzlich reduziert, sind zwei der Voraussetzungen für das erfolgreiche Bestehen eines Unternehmens – und dies ist letztlich das Ziel der Modellierung. In absehbarer Zeit ist es sicher möglich, auf geeignete Workflowmanagementsysteme zurückzugreifen, um dieses Ziel zu erreichen. Bei jedem einzelnen Modellierungsvorhaben gilt es aber stets aufs neue, die organisationale Struktur des Unternehmens zu beeinflussen, zu verändern. Dazu genügen dann technische Werkzeuge wie Workflowmanagementsysteme nicht mehr. Es gilt, soziotechnische Werkzeuge, Methoden, Verfahren, Vorgehensweisen zum Einsatz zu bringen.

Dazu ist die Entwicklung spezieller Methoden notwendig, die zielgerichtet eine menschen- und aufgabenzentrierte Modellierung unterstützen. Die Arbeit mit und an einem soziotechnischen System erfordert soziotechnische Verfahren. Sicherlich können beispielsweise die Methoden der partizipativen Systementwicklung und ihre Vorgehensmodelle wertvolle Hinweise geben und es kann auch auf solchen Vorgehensweisen aufgebaut werden. Dennoch sind hier aber spezielle Methoden gefragt, die den besonderen Anforderungen Rechnung tragen.

Dies ist die eigentliche Ausgangssituation der Modellierung soziotechnischer Systeme. Nicht erst *nach* einer abgeschlossenen Reorganisation für den neuen Betrieb Workflow-Anwendungen zu entwickeln, sondern beim Wandel mitzuhelfen. Daß dies nicht im Alleingang der Informatik möglich ist, sondern interdisziplinäre Zusammenarbeit mit Organisationsexperten, Arbeitswissenschaftlern und Ingenieuren unterschiedlichster Disziplinen erfordert, mag lediglich als Randaspekt erscheinen, ist aber von wesentlicher Bedeutung. Nicht dem Sog der scheinbaren Modellierbarkeit nachzugeben und nicht nur Datenmodellierung und Anwendungsentwicklung zu betreiben, sondern auch noch Arbeit, Workflows, Geschäftsprozesse und ganze Organisationen gestalten zu wollen, diese Beschränkung mag für manch einen als die wirkliche Herausforderung erscheinen. Die Modellierung – von der Datenmodellierung bis zum Workflowmanagement – muß sich der Erkenntnis öffnen, daß man nicht fast fertig ist und kurz vor der Entdeckung der allgemeinen Theorie der Modellierung steht, sondern sich erst am Anfang eines längeren Weges befindet und gerade erst einen Eindruck von der Gesamtproblematik gewonnen hat.

Nachbemerkung

Die Aussagen in diesem Beitrag mögen einigen Lesern zu wenig dokumentiert und belegt sein. Es fehlen Hinweise auf wissenschaftliche Untersuchungen, die sich detailliert mit der einen oder anderen Frage beschäftigt haben, Tendenzaussagen werden nicht durch einschlägige statistische Daten dokumentiert – dem ist sich der Autor durchaus bewußt. Dies

ist aber auch nicht die Aufgabe dieses Textes und kann sicherlich in der hier gebotenen Kürze auch nicht geleistet werden.

Der Beitrag soll vielmehr einer Meinung Ausdruck verleihen, die längst nicht mehr nur eine vereinzelte Sichtweise widerspiegelt – und dies haben sowohl die bisherigen Arbeiten in der EMISA-Arbeitsgruppe „Modellierung in soziotechnischen Systemen“, wie auch zahlreiche Gespräche auf Fachtagungen und Workshops gezeigt. Daher entfallen hier auch beispielsweise Literaturhinweise auf bekannte Fachpublikationen zugunsten neuer, in diesem Leserkreis vielleicht weniger bekannten Veröffentlichungen, die Anregungen zu neuen Überlegungen, neuen Ideen und neuen Lösungswegen sein sollen.

Der Beitrag soll kein resümierender Abschluß sein – im Gegenteil. Er soll vielmehr eine Diskussion in Gang setzen und will dazu anregen, (vielleicht schon viel zu) bekannte Themen aus neuer Perspektive zu betrachten und sich ihnen auf neuen Wegen zu nähern. Wer dies im Rahmen der EMISA-Arbeitsgruppe „Modellierung in soziotechnischen Systemen“ tun möchte, möge sich mit dem Autor postalisch (Institut Arbeit und Technik, Munscheidstraße 14, 45886 Gelsenkirchen) oder per EMail (paul@iatge.de) in Verbindung setzen. Das Ergebnisprotokoll der Gründungssitzung ist ebenfalls unter dieser Anschrift erhältlich.

Literatur

Böcker, 1995

Böcker, Heinz-Dieter (Hg.), 1995: Software-Ergonomie '95: Mensch – Computer – Interaktion: Anwendungsbereiche lernen voneinander. Stuttgart: B.G. Teubner.

Bolz / Kittler / Tholen, 1994

Bolz, Norbert / Kittler, Friedrich A. / Tholen, Christoph, 1994: Computer als Medium. München: Wilhelm Fink Verlag.

Brödner / Pekruhl, 1991

Brödner, Peter / Pekruhl, Ulrich, 1991: Rückkehr der Arbeit in die Fabrik. Wettbewerbsfähigkeit durch menschenzentrierte Erneuerung kundenorientierter Produktion. Gelsenkirchen: Institut Arbeit und Technik.

Jablonski, 1995

Jablonski, Stefan, 1995: Workflow-Management-Systeme: Motivation, Modellierung, Architektur. In: Informatik Spektrum (18/1). 13-24.

Küpper / Ortman, 1992

Küpper, Willi / Ortman, Günther, 1992: Mikropolitik: Rationalität, Macht und Spiele in Organisationen. Opladen: Westdeutscher Verlag.

Nake, 1986

Nake, Frieder, 1986: Die Verdoppelung des Werkzeugs. In: Rolf, Arno (Hg.), Neue Techniken alternativ. Hamburg: VSA-Verlag. 43-52.

Oberquelle, 1993

Oberquelle, Horst, 1993: Anpaßbarkeit von Groupware als Basis für die dynamische Gestaltung von computergestützter Gruppenarbeit. In: Konradt, Udo / Drisis, Leonidas (Hg.), Software-Ergonomie in der Gruppenarbeit. Opladen: Leske + Budrich. 37-54.

Oesterreich, 1981

Oesterreich, Rainer, 1981: Handlungsregulation und Kontrolle. München: Urban & Schwarzenberg.

Oesterreich, 1982

Oesterreich, Rainer, 1982: Der Begriff »Effizienz-Divergenz« als theoretischer Zugang zu Problemen der Planung des Handelns und seiner Motivation. In: Hacker, Winfried / Volpert, Walter / von Cranach, Mario (Hg.), Kognitive und motivationale Aspekte der Handlung. Wien: Huber. 110-122.

Ortmann et al., 1990

Ortmann, Günther / Windeler, Arnold / Becker, Albrecht / Schulz, Hans-Joachim, 1990: Computer und Macht in Organisationen – Mikropolitische Analysen. Opladen: Westdeutscher Verlag.

Ortmann, 1995

Ortmann, Günther, 1990: Formen der Produktion. Organisation und Rekursivität. Opladen: Westdeutscher Verlag.

Paul, 1995

Paul, Hansjürgen, 1995: Exploratives Agieren. Frankfurt/M.: Peter Lang.

Schelhowe, 1993

Schelhowe, Heidi, 1993: Werkzeug und Medium – Leitbilder in einem Projekt zur kooperativen Softwareentwicklung. In: Müller, Wilfried / Senghaas-Knobloch (Hg.), Arbeitsgerechte Softwaregestaltung – Leitbilder, Methoden, Beispiele. Münster: LIT-Verlag. 77-88.

Sinz, 1994

Sinz, Elmar, 1994: Geschäftsprozeßmodellierung als Grundlage für den Einsatz von Workflow-Management-Systemen. Tutorial 3 der D-CSCW 94 (Marburg). Bamberg: Otto-Friedrich-Universität Bamberg, Lehrstuhl für Wirtschaftsinformatik.

Sydow, 1992

Sydow, Jörg, 1992: Strategische Netzwerke: Evolution und Organisation. Wiesbaden: Gabler.

Versteegen, 1995

Versteegen, Gerhard, 1995: Schlanke Hälsa. In: iX. 86-92.

Wiborny, 1991

Wiborny, Werner, 1991: Datenmodellierung, Case-Management. Bonn: Addison-Wesley.

GI–Arbeitskreis

GRUNDLAGEN DER MODELLIERUNG UND AUSFÜHRUNG VON WORKFLOWS

Am 23. Juni 1995 fand die konstituierende Sitzung des Arbeitskreises "Grundlagen der Modellierung und Ausführung von Workflows" an der Universität Ulm statt. In Absprache mit den Fachgruppen 2.5.1 "Datenbanken" und 2.5.2 "EMISA", wird der Arbeitskreis direkt dem Fachausschuß 2.5 "Rechnergestützte Informationssysteme" zugeordnet.

Das Bestreben, einen Arbeitskreis zum Thema "Grundlagen der Modellierung und Ausführung von Workflows" zu gründen, wurde auf dem EMISA/MobIS–Fachgruppentreffen 1994 in Münster geweckt, als ein starkes Interesse der Fachgruppenbesucher an diesem Themengebiet festgestellt wurde. Vorbereitende Treffen fanden daran anschließend in Erlangen (24.11.1994) und in Karlsruhe (30.01.1995) statt. Zur Vorbereitung der Gründung dieses Arbeitskreises wurden von einer Arbeitsgruppe folgende Motivation und Aufgabenstellung erarbeitet:

Motivation:

Workflows sind abgrenzbare, meist arbeitsteilige Prozesse mit unterschiedlichem Komplexitätsgrad zur Erstellung oder Verwertung betrieblicher Leistungen. Aufgrund der Komplexität und Dynamik verschiedenster Anwendungsbereiche werden hohe Anforderungen an Workflowmodelle bzw. Workflow–Management–Systeme bzgl. Erweiterbarkeit, Anpaßbarkeit und Skalierbarkeit gestellt. Dies gilt insbesondere für die Modellierung und Ausführung von Geschäftsprozessen in verteilten, heterogenen Umgebungen.

Die Umsetzung der Anforderungen erfordert fundierte Kenntnisse im Bereich der Modellierung von Workflows und der Architektur von Workflow–Management–Systemen. Der Arbeitskreis will daher ein Forum für die Diskussion von grundlegenden Konzepten der Modellierung und Ausführung von Workflows bilden. Sowohl wissenschaftlich– als auch praxisorientierte Interessenten sind eingeladen, an den Aktivitäten des Arbeitskreises aktiv oder passiv teilzunehmen.

Aufgaben:

Die folgende Liste zeigt eine Auswahl fundamentaler Themenbereiche für den Arbeitskreis. Alle angeführten Einzelthemen sind beispielhaft zu verstehen:

1. Abgrenzung von Workflow–Management (u.a. Abgrenzung zu und Einordnung von CSCW, Groupware, Conferencing, Business Process Reengineering, transactional workflow, aktive Datenbanken)
2. Modellierung von Workflows (u.a. Definition eines implementierungsunabhängigen Modells unter Berücksichtigung von Kriterien wie Erweiterbarkeit, Offenheit, etc, Entwicklungsmethodik zur Modellbildung, Werkzeuge zur Analyse von modellierten Workflows (z.B. durch Simulation, Validierung))
3. Architektur von Workflow–Management–Systemen (u.a. Systemmodell, Ausführungsmodell, Verwendung von Komponenten der 'Middleware' (z.B. CORBA, DCE), Werkzeuge (z.B. zur Überwachung der Ausführung von Workflows und zur Optimierung))

Die Arbeiten des Arbeitskreises sollen sich auf Basisarbeit in einzelnen Arbeitsgruppen konzentrieren. In Abhängigkeit von den Interessen der aktiven Mitglieder des Arbeitskreises werden Arbeitsgruppen gebildet, die Themen aus den oben gezeigten Bereichen bearbeiten. Informelle Arbeitstreffen des Arbeitskreises dienen der Vorstellung und Diskussion der Ergebnisse der Arbeitsgruppen.

Neben der Durchführung dieser Basisarbeit setzt sich der Arbeitskreis zum Ziel, pro Jahr ein bis zwei Workshops oder Tagungen (mit) zu organisieren. Diese dienen als Forum zur allgemeinen, offenen Diskussion von Aspekten im Bereich der Modellierung und Ausführung von Workflows.

Obiger Vorschlag wurde von den Teilnehmern der konstituierenden Sitzung angenommen.

Zum Sprecher des Arbeitskreises "Grundlagen der Modellierung und Ausführung von Workflows" wurde Stefan Jablonski, Universität Erlangen-Nürnberg, gewählt. Stefan Wolf, LION GmbH Bochum, wurde zum stellvertretenden Sprecher gewählt. Als Publikationsorgane des Arbeitskreises sollen vorrangig die Publikationsorgane der Fachgruppen "Datenbanken" und "EMISA", der "Datenbank-Rundbrief" und das "EMISA-Forum" dienen.

Das nächste Treffen des Arbeitskreises findet am 26. September 1995 statt (Ort wird noch bestimmt). Ziel dieses Treffens wird sein, eine gemeinsame inhaltliche Arbeitsbasis für den Arbeitskreis zu entwickeln und erste Arbeitsgruppen zu konstituieren. Insbesondere soll ein gemeinsames Grundverständnis von Workflows, Workflow-Management und Workflow-Management-Systemen erarbeitet werden. Dies wird als notwendig erachtet, da auch innerhalb des Arbeitskreises vielfältige, teilweise widersprüchliche Interpretationen obiger Begriffe verwendet werden.

Nachfolgend finden Sie eine vorläufige Tagesordnung für das Treffen am 26. September 1995:

1. Begriffsklärung: Was ist ein Workflow?
 - Abgrenzung nach außen
 - Festlegung der fundamentalen Inhalte
2. Entwicklung eines gemeinsamen Arbeitsprogramms
3. Aufbau eines Glossars
4. Produktpräsentation (noch nicht bestätigt)

Wir möchten alle Interessenten an der Workflow-Thematik motivieren, im Arbeitskreis "Grundlagen der Modellierung und Ausführung von Workflows" mitzuarbeiten. Insbesondere möchten wir zum nächsten Treffen des Arbeitskreises am 26. September 1995 einladen.

Stefan Jablonski

Für Fragen und Anregungen stehen Ihnen immer gerne zur Verfügung:

Prof. Dr. Stefan Jablonski
Universität Erlangen-Nürnberg
Lehrstuhl für Datenbanksysteme
Martensstraße 3
D-91058 Erlangen
Tel. [49] (9131) 85-7885
Fax. [49] (9131) 32090
e-mail: jablonski@informatik.uni-erlangen.de

Dr. Stefan Wolf
LION Gesellschaft für
Systementwicklung mbH
Universitätsstr. 140
D-44799 Bochum
Tel. [49] 234 9709-381
Tel. [49] 234 9709-111
e-mail: wolf@lion.de

Tagungsbericht

Strategische Aspekte des Reengineering

Am 21. März 1995 fand an der Universität Münster der 3. Workshop der GI-Fachgruppe 5.1.3 "Reengineering und Wartung" statt. Er wurde organisiert von Dr. Knut Hildebrand und Dr. Stefan Eicker (beide Universität Münster). Den ca. 40 Teilnehmern wurden in den Vorträgen aktuelle Themen präsentiert; darüber hinaus wurde in der jeweils anschließenden Diskussion auf die Fragen der Zuhörer eingegangen.

M. Rezagholi (Siemens AG) trug vor zum Thema "Management wiederverwendungsorientierter Softwareerstellung". Er beleuchtete den gesamten Prozeß, der dazu nötig ist, unter den Aspekten Organisation und Technik. Ferner präsentierte er ein Vorgehensmodell für die Erstellung und Pflege von Assets, die verschiedene wiederverwendbare Objekte (Code, Domains usw.) enthalten können. Ein dediziertes Reifegradmodell diente der Identifikation des Standortes einer Organisation.

"Software-Sanierung als Alternative zur Neuentwicklung" war das Thema von O. Foshag (SES Software-Engineering-Service). Zu den Schwerpunkten seines Vortrags zählten: das Vorgehen unter Einsatz von Metriken und Tools, die Risiken und der Nutzen von Sanierungsprojekten, Analyse und Effizienz des Wartungsprozesses sowie die betriebswirtschaftliche Bewertung der Software.

R. Gastner (FhG-ISST) stellte ein "Verfahren zur Portfolio-Analyse für das Re-Engineering" vor. Es wird eingesetzt für strategische Entscheidungen, die durch das Management zu treffen sind. Auf der Basis von Kosten- und Nutzenschätzungen werden Vergleiche angestellt zwischen Reengineering, Standardsoftwareeinsatz, Neuentwicklung und Wartung, unter statischen und dynamischen Aspekten. Berücksichtigt werden ferner die Technologie und die Geschäftsprozesse.

A. Hofinger (IBM) erläuterte die "Bestandsanalyse als Grundlage der Software-Redevelopment-Strategie". Er legte das gesamte Vorgehen dar am Beispiel einer durchgeführten Analyse und ging ein auf die Punkte Prozesse, Bewertungsverfahren (mit 150 Kriterien), Fragestellungen, Auswertungsstrategie (Clustering) und RE-Kandidaten.

Im letzten Vortrag referierte K. Jantzen zum Thema "Grenzen des Reverse Engineering". Er differenzierte in Reverse Engineering-in-the-small auf der Implementierungsebene (Program Understanding) und Reverse Engineering-in-the-large hinsichtlich des anwendungsspezifischen Domain-Wissens. Entsprechend wird beim Vorgehen unterschieden zwischen der syntaktischen, semantischen und fachlichen Ebene.

Die Herausgabe eines Tagungsbandes ist geplant im Sommer dieses Jahres. Im anschließenden Fachgruppentreffen wurde ein weiterer Workshop für März 1996 angeregt. Thema wird die Wartung von Anwendungssystemen sein.

K. Hildebrand

Buchankündigung

G. Vossen, J. Becker (Hrsg.): *Geschäftsprozeßmodellierung und Workflow-Management — Modelle, Methoden, Werkzeuge*; International Thomson Publishing, Bonn, 1995, ISBN 3-8266-0124-6

Geschäftsprozeßmodellierung und Workflow-Management gehören zu den zentralen Anliegen der gegenwärtigen Forschung und Entwicklung im Bereich der rechnergestützten Informationssysteme. Beide Gebiete befinden sich jedoch derzeit keineswegs bereits in einem konsolidierten Zustand, der es erlauben würde, hier bereits von einem festen Satz verfügbarer Paradigmen und Techniken oder gar von Standard-Werkzeugen sprechen zu können, mit denen ein Anwender seine Probleme lösen kann. Statt dessen befinden sich beide Gebiete in einer Art Aufbruchstimmung, im Rahmen derer zur Zeit viel experimentiert und ausprobiert wird. Allerdings ist auch bereits eine Reihe von Teilerfolgen erzielt: Es gibt eine Vielzahl von *Modellen* zur Beschreibung und Strukturierung von Geschäftsprozessen und Workflows, es gibt eine Reihe von *Vorgehensweisen* zur Anwendung solcher Modelle auf konkrete betriebliche Gegebenheiten, und es gibt erste *Systeme*, welche einer Automatisierung sowohl der Geschäftsprozeßmodellierung als auch des Workflow-Managements dienen sollen.

Die Anregung zur Herausgabe eines Sammelbandes, in welchem eine Bestandsaufnahme der gegenwärtigen Situation versucht wird, hat das gemeinsame Fachgruppentreffen von EMISA und MobIS im Herbst 1994 in Münster gegeben. Allerdings stellt das Buch *keinen* Tagungsband dieses Fachgruppentreffens dar, sondern es beinhaltet eine Auswahl von Arbeiten, über die in Münster vorgetragen wurde, und es wurden sogar Arbeiten aufgenommen von Autoren, die am Fachgruppentreffen nicht beteiligt waren.

Die Auswahl des hier beschriebenen Stoffes erfolgte unter verschiedenen Gesichtspunkten: Einerseits stößt die Problematik der Modellierung von Geschäftsprozessen und des Workflow-Managements sowohl in der Informatik als auch in der Wirtschaftsinformatik auf zunehmendes Interesse. Allein aus der Tatsache, daß diese beiden Gebiete die Problematik in unterschiedlicher Weise aufgreifen und bearbeiten, ergibt sich ein interessantes Spannungsfeld, was im vorliegenden Text (an dem Vertreter beider Gebiete mitgewirkt haben) seinen Niederschlag findet. Wichtig erscheint uns andererseits, daß der Leser sowohl die verschiedenen möglichen Vorgehensweisen, etwa bei der Analyse und der Modellierung von Geschäftsprozessen, als auch derzeit als typisch anzusehende Anwendungen kennenlernt.

Inhaltsübersicht:

1. Geschäftsprozeßmodellierung und Workflow-Management: Eine Einführung (J. Becker, G. Vossen)
- I. Modellierung und Analyse von Geschäftsprozessen**
2. Geschäftsprozeßmodellierung innerhalb einer Unternehmensarchitektur (A.-W. Scheer, W. Jost)
3. Geschäftsprozeßmodellierung im Rahmen des Semantischen Objektmodells (O.K. Ferstl, E.J. Sinz)

4. Objektorientierte Analysekonzepte in der Unternehmensmodellierung (C. Kohl)
5. Das Unternehmensmodell als Basis der Wiederverwendung bei der Geschäftsprozeßmodellierung (M. Bertram)
6. Ein Referenzschema zur Organisationbeschreibung (A. Winter, J. Ebert)
7. Von der informalen zur formalen Geschäftsprozeßmodellierung (P. Elgass, H. Krcmar, A. Oberweis)
8. Geschäftsprozeßmodellierung mit INCOME (P. Jaeschke)
9. Geschäftsprozeßadäquate Informationssystemadaption durch generische Strukturen (P. Loos)
10. Ein Ansatz zur Optimierung von Geschäftsprozessen (A. Taudes, P. Cilek, M. Natter)

II. Anwendungen der Geschäftsprozeßmodellierung

11. Effiziente Prozeßgestaltung im Rechnungswesen (M. Rosemann, R. Schulte)
12. Prozeßorientierung in der Produktionsplanung und -steuerung (K. Turowski)
13. Fertigungsprozesse und Fertigungssteuerung (W. Dangelmaier, W. Felser)
14. Gefahrstoffe und Sondermüll in der betrieblichen Logistik (R. Oetinger)
15. Prozeßorientierung in Handelsunternehmen (R. Schütte)

III. Workflows und Workflow-Management

16. Klassifikation von Vorgangsverwaltungssystemen (W. Schulze, M. Boehm)
17. Koordinationsmodelle und Softwarearchitekturen als Basis für die Auswahl und Spezialisierung von Workflow Management Systemen (K. Schwab)
18. Vorgangsmanagementsysteme: Modellierungs- und Implementierungskonzepte (T. Rose)
19. Transaktionskonzepte für Workflow-Managementsysteme (F. Leymann)
20. Verifikation von Workflows in IBM FlowMark (D. Roller)
21. Die Architektur des modularen Workflow-Management-Systems MOBILE (C. Bußler, S. Jablonski)
22. Ein Workflow-Managementsystem auf der Basis aktiver Datenbanken (J. Eder, H. Groiss)
23. Entwicklung Workflow-Management-geeigneter Software-Systeme (G. Dinkhoff, V. Gruhn)

Literaturverzeichnis

Das Buch wird etwa 350 Seiten umfassen und voraussichtlich ab Ende August 1995 im Buchhandel erhältlich sein. Es ist auch elektronisch direkt beim Verlag bestellbar über

<http://www.ora.de>

Münster, im Juni 1995

G. Vossen

Ankündigungen

(Für Hinweise auf EMISA-Veranstaltungen und Veranstaltungen mit EMISA-Beteiligung siehe auch Seite 5-17!)

- "21st Intl. Conference on Very Large Data Bases VLDB",
11.-15. September 1995 in Zürich (Kontakt: vldb95@ifi.unizh.ch)
- GI-FG 2.5.1 / GI-FG 2.5.2 / SI-FG DBTA Fachgespräch
"Entwurf und Entwicklung verteilter Informationssysteme"
im Rahmen der gemeinsamen Jahrestagung der GI und der SI,
18. September 1995 in Zürich
- Fachgespräch "Informationsverarbeitung und Kommunikation im Gesundheitswesen:
Planung und Bewertung von Krankenhausinformationssystemen"
im Rahmen der gemeinsamen Jahrestagung der GI und der SI,
20. September 1995 in Zürich
- 6th Int. Workshop on "Foundations of Models and Languages for Data and Object:
Specification and Semantics of Databases and Database Languages",
25.-29. September 1995 in Dettingen
- "Fifth European Software Engineering Conference ESEC'95",
25.-28. September 1995 in Sitges/Spanien
- Int. Symposium "Informatik für den Umweltschutz",
27.-29. September 1995 in Berlin
- Fachtagung "Geschäftsprozesse und Workflow-Systeme in der evolutionären Unter-
nehmung" der GI-Fachgruppe MobIS,
5.-6. Oktober 1995 in Bamberg
- Treffen der GI-Arbeitsgruppe "Petrietze und Informationssysteme in der Praxis",
9. Oktober 1995 in Bonn
- Workshop "Algorithmen und Werkzeuge für Petrietze",
10.-11. Oktober 1995 in Oldenburg
- EMISA-Mitgliederversammlung,
12. Oktober 1995 in Karlsruhe

- EMISA-Fachgruppentreffen "Requirements Engineering für Informationssysteme",
12. - 13. Oktober 1995 in Karlsruhe
- EMISA-Workshop "Natürlichsprachliche Entwicklung von Informationssystemen",
28.-30. Mai 1996 in Tutzing
- Tenth European Conference on "Object-Oriented Programming ECOOP96",
8.-12. Juli 1996 in Linz/Österreich
- EMISA-Fachgruppentreffen "Informationsserver für das Internet: Anforderungen,
Konzepte, Methoden",
10.-11. Oktober 1996 in Aachen

Tagungsort

Universität Bamberg
Standort Feldkirchenstraße
Feldkirchenstr. 21
D-96045 Bamberg

Alle Vorträge finden in Hörsälen des Standortes Feldkirchenstraße der Universität Bamberg statt.

Anreisemöglichkeiten bestehen mit dem Auto (A3/B505, A70, A73) oder mit der Bahn (Intercity).

Der Standort Feldkirchenstraße ist vom Hauptbahnhof mit den Buslinien 1 bzw. 14 (Richtung Gartenstadt, Haltestelle Kloster-Banz-Straße) oder der Buslinie 7 (Richtung Memmelsdorf, Haltestelle Feldkirchenstraße) erreichbar.

Am Universitätsgelände stehen Parkmöglichkeiten in begrenzter Anzahl zur Verfügung.

Tagungsbüro

Tel.: ++49 951 863-2512

Fax: ++49 951 39636

e-mail: mobis95@seda.sowi.uni-bamberg.de

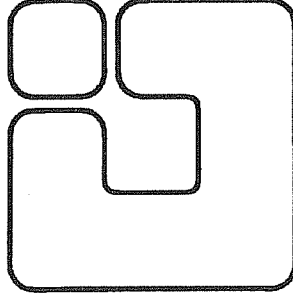
WWW: <http://www.seda.sowi.uni-bamberg.de/~fa52/>

Tagungsleitung

Prof. Dr. Elmar J. Sinz, Heiko Raue,
Universität Bamberg

Veranstalter

GI-Fachausschuß 5.2 * GI-Fachgruppe 5.2.1
"Informationssystem-Architekturen" MobIS "Modellierung betrieblicher
Informationssysteme"



Fachbereich 5 Wirtschaftsinformatik

Fachgruppe 5.2.1

Modellierung betrieblicher Informationssysteme
(MobIS)

Fachtagung

*Geschäftsprozesse und Workflow-Systeme in der
evolutionären Unternehmung*

Einladung und Programm

5. und 6. Oktober 1995
Universität Bamberg

Tagungsprogramm

Methodische Fragen der Architektur von Informationssystemen (IS) in Wirtschaft und Verwaltung stellen eine permanente Anforderung an die Wirtschaftsinformatik dar. Diese Informationssysteme umfassen sowohl die betriebliche Organisation als auch die Gesamtheit der computergestützten Anwendungssysteme. Durch geeignete IS-Architekturen werden die Voraussetzungen für flexible, evolutiv entwickelbare Informationssysteme geschaffen. Die Gestaltung der IS-Architekturen ist an den Unternehmenszielen ausgerichtet.

Geschäftsprozesse stellen Lösungsverfahren für die Umsetzung der Unternehmensziele dar. Voraussetzungen für eine evolutionäre Fortentwicklung der Unternehmung sind flexibel anpaßbare Geschäftsprozesse sowie deren geeignete Unterstützung durch Workflow-Systeme. Die Tagungsbeiträge lassen sich vor diesem Hintergrund folgenden Schwerpunkten zuordnen:

- Modellierung von Geschäftsprozessen inkl. Referenzmodelle,
- Unterstützung von Geschäftsprozessen durch Workflow-Systeme und
- workflow-orientierte Architekturen von Informationssystemen bzw. Anwendungssystemen.

Das Tagungsprogramm wird durch Beiträge aus Bamberger Forschungsprojekten abgerundet, die sich mit den Schwerpunktthemen der Tagung befassen.

Veröffentlichung

Die Vortragskurzfassungen zur Tagung werden im FA-Rundbrief "Informationssystem-Architekturen" veröffentlicht.

Tagungsprogramm

Donnerstag, 5. Oktober 1995

09.00-09.15 Begrüßung (Hörsaal A)

09.15-10.00 Sitzung 1: Einführung

Hörsaal A
I. Sinz, E.J., Universität Bamberg: Geschäftsprozesse und Workflow-Systeme - Modelle und Architekturen

10.00-10.30 Kaffeepause

10.30-12.45 Sitzung 2: Geschäftsprozeßmodellierung I

Hörsaal A
2. Ferstl, O.K., Universität Bamberg: Lenkung industrieller Geschäftsprozesse

3. Strobel, M., Boston Consulting Group: Simultane Gestaltung verteilter Anwendungssysteme und der betrieblichen Organisation auf der Grundlage von Geschäftsprozeßmodellen

4. Neuscheler, F., Sternemann, K.H., Fachhochschule Offenburg: Business Process (Re-) Engineering with CIMOSA at a Paper Mill

12.45-14.00 Mittagspause

14.00-15.30 Sitzung 3: Workflow-orientierte Architekturen I

Hörsaal A
5. Derszteler, G., Technische Universität Berlin: Workflow Dictionary - Ein Ansatz zur Integration von Organisationsmodellierung und Workflow Management

6. Gerstner, R., SAP AG: Ereignisse als Basis einer workfloworientierten Architektur von Anwendungssystemen

15.30-16.00 Kaffeepause

16.00-17.30 Sitzung 4: Unterstützung von GP durch WF-Systeme I

Hörsaal A
7. Pohl, K., Jarke, M., Dörmges, R., RWTH Aachen: Unterstützung schwach strukturierter Geschäftsprozesse

8. Böhne, A.J., Humboldt-Universität zu Berlin: Wirtschaftlichkeit prozessorientierter Informationssysteme - Ein Ansatz zur Modellierung von Kontextänderungen in Geschäftsprozessen

Hörsaal B
Sitzung 5: Bamberger Forschungsprojekte I

9. Raue, H., Universität Bamberg: Wiederverwendung in Geschäftsprozeßmodellen

10. Krumbiegel, J., Universität Bamberg: Modellierung von Universitätsprozessen

17.30-18.00 Kaffeepause

18.00-19.00 Mitgliederversammlung '95 der FG 5.2.1 MobIS (Hörsaal A)

Tagungsprogramm

Freitag, 6. Oktober 1995

- 8.30-10.00**
Hörsaal A **Sitzung 6: Unterstützung von GP durch WF-Systeme II**
11. Conen, W., Lenke, C. Universität Essen, Krukis, R., IPT Karlsruhe:
Workflow-Management-Systeme - Integrierte Modellierung und
Ausführung flexibler Prozesse
12. Wächter, H., SAP AG: Flexible Geschäftsprozesse mit SAP Business Workflow 3.0
- 10.00-10.30**
Kaffeepause
- 10.30-12.00**
Hörsaal A **Sitzung 7: Workflow-orientierte Architekturen II**
13. Amberg, M., Universität Bamberg: Ableitung von WF-Spezifikationen für Workflow-Managementsysteme aus Geschäftsprozessmodellen des SOM-Ansatzes
14. Ruppelta, W., SNI AG: Flexible Geschäftsprozesse mit Workflow-Anwendungen
- Hörsaal B* **Sitzung 8: Bamberger Forschungsprojekte II**
15. Zimmermann, F.O., Universität Bamberg: Ein Geschäftsprozessmodell des Controlling
16. Hazebrouck, J.-P., Universität Bamberg: Führungsinformationssysteme als Mittel zur Lenkung industrieller Geschäftsprozesse
- 12.00-13.15**
Mittagspause
- 13.15-15.30**
Hörsaal A **Sitzung 9: Geschäftsprozessmodellierung II**
17. Berkau, C., IDS Prof. Scheer: Modellbasiertes Controlling für komplexe Geschäftsprozesse
18. Rump, F., Appelrath, H.-J., Universität Oldenburg: Ereignisgesteuerte Prozessketten zur formal fundierten Geschäftsprozessmodellierung
19. Kraatz, N., Rose, T., Kittel, I., FAW Ulm: Flexible Geschäftsprozessmodellierung mit ProMAX
- 15.30-15.45**
Schlußwort (Hörsaal A)

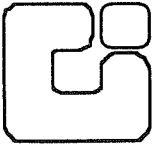
Bitte an folgende Adresse senden:

Tagungsbüro MobIS 1995
Prof. Dr. Elmar J. Sinz
Lehrstuhl für Wirtschaftsinformatik
Feldkirchenstr. 21
D-96045 Bamberg



Bitte an folgende Adresse senden:

Fremdenverkehrsamt
der Stadt Bamberg
Postfach 120 163
D-96033 Bamberg



Deutsche Gesellschaft für Medizinische Informatik, Biometrie
und Epidemiologie e.V. - Fachbereich 'Medizinische Informatik'

Gesellschaft für Informatik e.V.
Fachausschuß 4.7 'Medizinische Informatik'



Methoden und Werkzeuge für das Management von Krankenhausinformationssystemen

Arbeitskreis des GI-Fachausschusses 4.7 'Medizinische Informatik'
Projektgruppe in dem Fachbereich 'Medizinische Informatik' der GMDS

Leitung: Alfred Winter (Heidelberg)
stv. Leitung: Roland Zimmerling (Oldenburg)

- Aufruf zur Mitarbeit -

Das systematische Management von Krankenhausinformationssystemen ist, wie bei anderen betrieblichen Informationssystemen auch, notwendig, damit eine angemessene Durchführung der in einem Krankenhaus erforderlichen informationsverarbeitenden Verfahren unter Berücksichtigung einer wirtschaftlichen Betriebsführung ermöglicht wird. Management von Krankenhausinformationssystemen umfaßt die

- Gestaltung (d.h. sowohl den Aufbau als auch die Weiterführung) und die
- Überwachung

von Krankenhausinformationssystemen. Dabei sind die branchenspezifischen Belange von Krankenhäusern zu berücksichtigen, um einen angemessenen Betrieb der Krankenhausinformationssysteme und damit eine adäquate Informationsverarbeitung für die Patientenversorgung und - im universitären Bereich - für die medizinische Forschung und Lehre zu erzielen.

Dieses Ziel kann nur durch eine systematische Vorgehensweise erreicht werden. Die hierzu erforderliche (Weiter-) Entwicklung von Methoden und Werkzeugen für das systematische Management von Krankenhausinformationssystemen umfaßt auch die Entwicklung von Referenzmodellen für Krankenhausinformationssysteme und ist eine anspruchsvolle Aufgabe sowohl für die Informatik als auch für die Medizinische Informatik; ein interdisziplinärer Erfahrungs- und Wissensaustausch zwischen Informatik, Medizinischer Informatik und Wirtschaftsinformatik bildet die Voraussetzung.

Zur Intensivierung dieses Erfahrungs- und Wissensaustauschs wurde innerhalb der Gesellschaft für Informatik (GI) der Arbeitskreis 'Methoden und Werkzeuge für das Management von Krankenhausinformationssystemen' des Fachausschusses 4.7 eingerichtet, der gleichzeitig Projektgruppe im Fachbereich Medizinische Informatik der Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie (GMDS) ist. Der Arbeitskreis soll ein aktives Bindeglied für die entsprechenden Gliederungen von GI und GMDS sein.

Sollten Sie sich für eine Mitarbeit in dem Arbeitskreis bzw. der Projektgruppe interessieren, setzen Sie sich bitte mit dem Leiter oder stv. Leiter in Verbindung. Es wäre hilfreich, wenn Sie bei dieser Gelegenheit Ihre eigenen Interessen und Vorstellungen kurz skizzieren könnten. Bitte vergessen Sie auch nicht die Angabe Ihrer e-mail-Adresse (falls vorhanden).

PD Dr. Alfred Winter
Klinikum der Universität Heidelberg
Abt. Medizinische Informatik
Im Neuenheimer Feld 400
69120 Heidelberg
Tel.: 06221/56-7497, -7483
e-mail: Dr._Alfred_Winter@krzmail.krz.uni-
heidelberg.de

Dr. Roland Zimmerling
Kuratorium OFFIS e.V.
FB 9 Betriebswirtschaftliche Informatik
Westerstraße 10-12
26121 Oldenburg
Tel.: 0441/9707-4, -393
e-mail: zimmerli@hera.offis.uni-oldenburg.de

7. Juni 1995

Pressemitteilung 1/95

Computer machen die Natur verständlich

"Informatik für den Umweltschutz" / Internationales Symposium in Berlin (27.-29.9.95)

Natur und Computer stellen im modernen Umweltschutz keine Gegensätze mehr dar. Ohne Methoden und Geräte der Informationstechnik wäre unser Wissen über die Umwelt um vieles ärmer. "Unsichtbare" Prozesse wie die Entstehung des Ozonlochs über der Antarktis, die Wanderung von Schadstoffen im Grundwasser oder die Turbulenzen des künftigen Weltklimas werden erst durch die Auswertung zahlreicher Meßdaten in Großrechnern und PC darstellbar und verständlich.

Auch die Informatik mit ihren Methoden und Techniken wie Bildverarbeitung, Rechnernetze, Daten- und Methodenbanken, Expertensysteme, Grafik, Modellbildung oder Simulation kommt bei der Informationsverarbeitung im Umweltsektor eine zentrale Rolle zu.

Gerade die räumliche Differenzierung und zeitliche Veränderungen von Ökosystemen lassen sich mit Hilfe leistungsfähiger Informationstechniken immer realitätsgreuer abbilden und simulieren. "Raum und Zeit in Umweltinformationssystemen" bildet daher das Rahmenthema des 9. Internationalen Symposiums "Informatik für den Umweltschutz - Computer Science for Environment Protection" (CSEP), das vom 27. bis 29. September 1995 in Berlin stattfindet.

Veranstaltet wird die Tagung, zu der rund 600 Experten erwartet werden, von der Gesellschaft für Informatik e.V. (GI) und dem Fortbildungszentrum Gesundheits- und Umweltschutz Berlin e.V. (FGU BERLIN). Die Teilnehmer sind Wissenschaftler, Entwickler und Anwender von Umweltinformationssystemen aus Informatik, Ökologie und Geowissenschaften in Forschung, Verwaltung und betrieblicher Praxis.

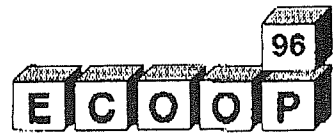
Mit dem Schwerpunktthema "Raum und Zeit" widmet sich die Tagung insbesondere den anwendungs-integrierten Geoinformationssystemen und der Modellierung dynamischer Vorgänge. Vorgestellt und diskutiert werden unter anderem Großprojekte des internationalen Umweltschutzes (z.B. in Zentral-Asien und Osteuropa) mit dem Aufbau und der Auswertung komplexer Umweltinformationssysteme, die Verbindung der Umweltinformatik mit dem Gesundheitsschutz (Belastung von Nahrungsmitteln, Risikoabschätzung, "Gesundheitsatlas Europa"), sowie Anwendungen im kommunalen Bereich. Das Fachgespräch "Umweltinformationssysteme in der Produktion" besitzt mit der neuen Europäischen Öko-Audit-Verordnung einen besonders aktuellen Hintergrund. Die Berliner Diskussion will die Reallisierungschancen betrieblicher Umweltinformationssysteme auf der Basis innovativer Informatik-Methoden aufzeigen.

Mehr als 100 Referenten werden über ihre Forschungs- und Praxisprojekte berichten. Zahlreiche Hersteller, Entwickler, Forschungseinrichtungen und Behörden werden im Rahmen der begleitenden Fachausstellung innovative Umweltinformationssysteme sowie praktische Ergebnisse präsentieren.

Nähere Informationen:

Horst Kremers, Gesellschaft für Informatik
(Tagungsleiter)
Postfach 200548
13515 Berlin
Telefon/Fax: 030-372 85 87

FGU BERLIN
Kleiststraße 23-26
10787 Berlin
Tel: 030-21 295 411 / 412
Fax: 030-21 295 420



Tenth European Conference on
Object-Oriented Programming
July 8-12, 1996 Linz - Austria

CALL FOR CONTRIBUTIONS

ECOOP '96 is the Tenth European Conference on Object-Oriented Programming. The conference aims at bringing together researchers and practitioners from academia and industry to discuss and exchange new developments in object-oriented languages, systems and methods.

CALL FOR PAPERS

Authors are invited to submit either original research contributions, or experience reports that provide new insight into the use of object-oriented and related techniques. Suggested topics include, but are not limited to:

- Languages and their implementations
- Combination of OOP with other language paradigms
- OO databases and object persistence
- Concurrency and distribution
- Multiagent systems
- Theoretical foundations
- Formal methods
- OO analysis, design and metrics
- Class management and class evolution
- Reuse, components and frameworks
- Applications and experiences
- Teaching OO concepts and methods

For Proposals for TUTORIALS, WORKSHOPS, PANELS, EXHIBITS, and DEMONSTRATIONS & POSTERS, please, contact the appropriate chairs and homepages via WWW:

<http://www.ifs.uni-linz.ac.at/ecoop96>

Conference Co-Chairs
Oscar Nierstrasz (University of Berne, CH)
Peter Wegner (Brown University, USA)

Programme Chair
Pierre Cointe
Department d'Informatique
Ecole des Mines de Nantes
E-mail: ecoop96@emn.fr

Tutorial Chair
Hanspeter Mössenböck
Department of System Software
University of Linz
E-mail: moessenboeck@ssw.uni-linz.ac.at

Workshop Chair
Max Mühlhölzer
University of Linz - Telecooperation
E-mail: max@tk.uni-linz.ac.at

Panel Chair
Markku Sakkinen
Department of Information Systems
University of Linz
E-mail: markku@ifs.uni-linz.ac.at

Exhibits and Demonstrations Chair
Wolfgang Pree
University of Linz - Software Engineering
E-mail: pree@swe.uni-linz.ac.at

DEADLINES FOR CALL FOR PAPERS

Submission of Papers: Dec. 1, 1995
Notification of acceptance: Feb. 15, 1996
Camera-ready copies: April 1, 1996

GENERAL INFORMATION

Gerti Kappel
ECOOP '96 Organizing Chair
Department of Information Systems
University of Linz
Altenbergerstr. 69
A-4040 Linz, Austria
Tel.: +43-732-2468-880
Fax: +43-732-2468-9308
E-mail: ecoop96@ifs.uni-linz.ac.at

Programme Committee

Mehmet Aksit (University of Twente, NL)
Giuseppe Attardi (University of Pisa, I)
Francois Bancillon (O2 Technology, F)
Yves Caseau (Bouygues, F)
Pierre Cointe (Ecole des Mines de Nantes, F; Chair)
Theo D'Hondt (Brussels Free University, B)
Erich Gamma (Taligent, USA)
Rachid Guerraoui (Ecole Poly. Federale de Lausanne, CH)
Ivar Jacobson (Objectory AB, S)
Mehdi Jazayeri (Technical University of Vienna, A)
Gregor Kiczales (Xerox Parc, USA)
Karl Lieberherr (Northeastern University, USA)
Ole Lehrmann Madsen (Aarhus University, DK)
Tom Maibaum (Imperial College, UK)
Boris Magnusson (Lund University, S)
Jacques Malenfant (University of Montreal, CDN)
Bertrand Meyer (ISE, USA+P)
Jose Meseguer (SRI, USA)
Max Mühlhölzer (University of Linz, A)
Walter Olthoff (DPKI GmbH, D)
OOPSLA '96 programme chair (requested)
Jens Palsberg (MIT, USA)
Markku Sakkinen (University of Jyväskylä, SF)
Dave Thomas (OTI Inc, CDN)
Mario Tokoro (Keio U/ Sony CSL, J)
Aldnor Yonezawa (Tokyo University, J)
Roberto Zicari (J.W.Goethe-University, D)



Location:

The conference takes place at the campus of the Johannes Kepler University of Linz, located at the Danube, in the heart of Austria right between Vienna and Salzburg. Linz has around 200.000 residents and is surrounded by the gentle hills of the Bohemian landscape to the north, and by the starting Alps to the west. The famous Salzkammergut with its lakes and hills is starting to the west of Linz. Linz has an own airport with direct flights to most cities in Germany and Switzerland, and connections to all destinations worldwide with stopover in Vienna.

Fachexperten im EMISA Leitungsgremium:

Wirtschaftsinformatik/FG 5.2.1:

Dipl.-Wi.-Inf. Heiko Raue
Universität Bamberg
Lehrstuhl für Wirtschaftsinformatik
Feldkirchenstr. 21
D 96045 Bamberg
Tel. 0951-863-2514
Fax 0951-39636
Email raue@seda.sowi.uni-bamberg.de

Petri-Netze:

Prof. Dr. Gert Scheschonk
C.I.T. Communication and Information
Technology GmbH
Ackerstraße 71-76
D 13355 Berlin
Tel. 030-4636077
Fax 030-4641610

Knowledge Engineering:

Prof. Dr. Rudi Studer
Universität Karlsruhe
Institut für Angewandte Informatik und
Formale Beschreibungsverfahren
Kollegiengebäude am Ehrenhof
D 76128 Karlsruhe
Tel. 0721-608-3923
Fax 0721-693717
Email studer@aifb.uni-karlsruhe.de

EMISA Leitungsgremium

| | |
|--|--|
| <p>Prof. Dr. Gottfried Vossen (Sprecher des Leitungsgremiums) Universität Münster Institut für Wirtschaftsinformatik Grevererstr. 91</p> <p>D 48159 Münster Tel. 0251-9275-103 Fax 0251-83-9754 Email vossen@uni-muenster.de</p> | <p>Dr. Andreas Oberweis (Redaktion EMISA FORUM) Universität Karlsruhe Institut für Angewandte Informatik und Formale Beschreibungsverfahren Kollegiengebäude am Ehrenhof D 76128 Karlsruhe Tel. 0721-608-4283 Fax 0721-693717 Email oberweis@aifb.uni-karlsruhe.de</p> |
| <p>Prof. Dr. Stefan Jablonski</p> <p>Universität Erlangen-Nürnberg Informatik VI Martensstr. 3 D 91058 Erlangen Tel. 09131-857885 Fax 09131-32090 Email jablonski@informatik.uni-erlangen.de</p> | <p>Dr. Heinrich Jasper</p> <p>Universität Oldenburg Fachbereich Informatik Postfach 2503 D 26111 Oldenburg Tel. 0441-798-2995 Fax 0441-798-2971 Email jasper@informatik.uni-oldenburg.de</p> |
| <p>Dr. Manfred Jeusfeld</p> <p>Hong Kong University of Science and Technology ISMT Sai Kung Hong Kong</p> <p>Email jeusfeld@usthk.ust.hk</p> | <p>Dr. Roland Kaschek</p> <p>Universität Klagenfurt Institut für Informatik Universitätsstr. 67 A 9022 Klagenfurt Tel. +43-463-2700-527 Fax +43-463-2700-505 Email kaschek@ifi.uni-klu.ac.at</p> |
| <p>Prof. Dr. Udo Lipeck</p> <p>Universität Hannover Institut für Informatik Lange Laube 22 D 30159 Hannover Tel. 0511-762-4951 Fax 0511-762-4961 Email ul@informatik.uni-hannover.de</p> | <p>Dr. Perdita Lühr</p> <p>Quantum GmbH Emil-Figge-Str. 83 D 44227 Dortmund Tel. 0231-9749-171 Fax 0231-9749-3 Email loehr@quantum.de</p> |
| <p>Prof. Dr. Erich Ortner</p> <p>Universität Konstanz Lehrstuhl für Informationsmanagement Postfach 5560 D 78434 Konstanz Tel. 07531-88-2976 Fax 07531-88-2601 Email ortner@inf-wiss.ivp.uni-konstanz.de</p> | <p>Dipl.-Inform. Helmut Thoma</p> <p>CIBA-GEIGY AG Information Technology R-1008.8.05 CH 4002 Basel Tel. +41-61-6973419 Fax +41-61-6972469 Email helmut.thoma@chbs.mhs.ciba.com</p> |

REDAKTIONSSCHLUSS FÜR

DIE NÄCHSTE AUSGABE

EMISA FORUM HEFT 1/1996:

5. DEZEMBER 1995