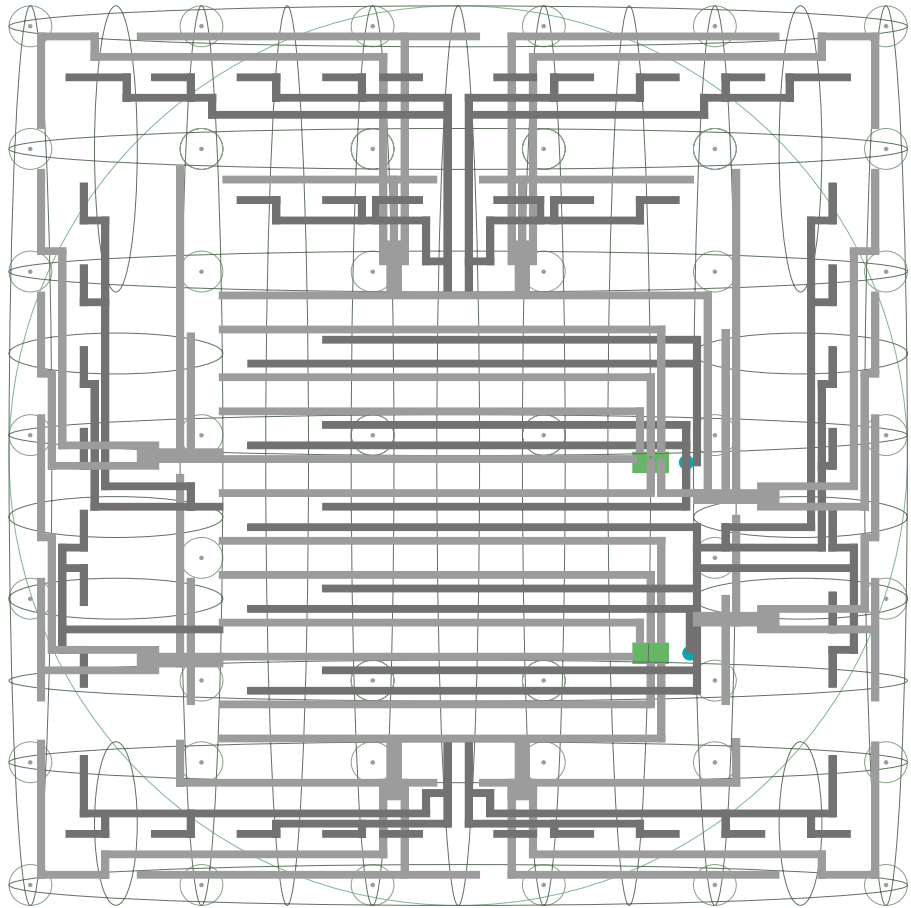


# a4 - digitales bauen

*Ein Modell für die weitgehende  
Computerunterstützung von Entwurf,  
Konstruktion und Betrieb  
von Gebäuden.*



Ludger Hovestadt  
Bruchwaldstr. 36  
76229 Karlsruhe

phone #49 721 608 2167  
fax #49 721 66 11 15  
email [ludger@ifib1.ifib.uni-karlsruhe.de](mailto:ludger@ifib1.ifib.uni-karlsruhe.de)

Ich danke Herrn Haller, Herrn Flemming und Herrn  
Gauchel für Ihre Unterstützung dieser Arbeit.

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>1</b>
	MIDI/ARMILLA 1	
	Objekte und Regeln 2	
	Blackboard System 2	
	Multiuser Blackboard System 3	
	Der vieldimensionale Datenraum 3	
<b>2</b>	<b>Das Problem</b>	<b>5</b>
	Der Bauprozeß 5	
	one-of-a-kind 5	
	Die heutige Arbeitsweise 5	
	Ausweitung der Planungsaufgaben 6	
	Intelligente Gebäude 6	
	Spezialisierung und Diversifizierung 6	
	Internationalisierung 7	
	Dynamisierung 7	
	Informatisierung 7	
	Computeranwendungen 8	
	Spezialisierung 8	
	Künstliche Intelligenz 8	
	Datenmenge 9	
	Nutzerkontrolle 9	
	Vernetzung 9	
	Multimedia 9	
	Programmierung durch Anwender 9	
	Integrierende Strukturen 10	
	Klassifikationen 10	
	Relationale Datenbanken zur Gebäudeverwaltung 10	
	Gebäudemodelle 11	
	Formalisierungen des Entwerfens 12	
	Baukästen 13	
	Gebäudetypologien 14	
	Anforderungen an ein Integrationsmodell 14	
<b>3</b>	<b>Das A4-Modell</b>	<b>15</b>
	Einleitung 15	
	Die Container 16	
	Der Datenraum 16	
	Die Dimensionen x, y und z 17	
	Die Dimension Zeit 21	
	Die Dimension Größenordnung 22	
	Die Dimension Auflösung 26	
	Die Dimension Teilsystem 28	
	Die Dimension Morphologie 29	

	Die Dimension Alternativen	30
	Die Dimension Nutzer	31
	Die Dimension Timetag	31
	Die Dimension Komposition	32
	Die Dimension Meta	32
	Die Skalen	33
<b>4</b>	<b>Die Navigation im A4-Modell</b>	<b>35</b>
<b>5</b>	<b>Methoden der KI im A4-Modell</b>	<b>39</b>
	Die Wissensarten	39
	Fallvergleichendes Wissen	40
	Heuristisches Wissen	40
	Statistisches (probabilistisches) Wissen	42
	Modellbasiertes (kausales) Wissen	42
	Die Problemlöseverfahren	43
	Klassifikation (Diagnostik)	43
	Konstruktionsmethoden	44
	Simulationsmethoden	46
	Die Wissensrepräsentationen	47
	Frames	47
	Regeln	48
	Constraints	48
	Truth Maintenance System	49
<b>6</b>	<b>Datenbanken und das A4-Modell</b>	<b>50</b>
	Passive Objekte	50
	Aktive Objekte	52
	WAVE	53
	Blackboard Systeme	54
	Die Koordination im A4-Modell	55
<b>7</b>	<b>Interfaces und das A4-Modell</b>	<b>60</b>
	komplexe Interfaces	60
	naive Interfaces	60
	Allround Systeme	61
<b>8</b>	<b>Zusammenfassung, Ausblick</b>	<b>62</b>
<b>9</b>	<b>Anhang</b>	<b>63</b>
	Anhang I: Der Forschungskontext	63
	Anhang II: armilla	64
	Anhang III: Video	65
	Anhang IV: Implementationen	65
	A1	65
	A2	66
	A3	66
	A4 (UNIX - AutoCAD - KnowledgeCraft)	66
	A4 (Objective_C, NeXTSTEP)	67

<b>10</b>	<b>Glossar</b>	<b>68</b>
<b>11</b>	<b>Literatur</b>	<b>72</b>
<b>12</b>	<b>Autoren</b>	<b>80</b>
<b>13</b>	<b>Index</b>	<b>81</b>

# 1 Einleitung

## MIDI/ARMILLA



**Bild 1** Der Gebäudebaukasten MIDI für bis zu viergeschossige, hochinstallierte Gebäude, wie Büro-, Schul- oder Laborgebäude. Er integriert alle technischen Systeme, so daß die Gebäude leicht an die sich ändernden Anforderungen angepaßt werden können. Abgebildet ist das Ausbildungszentrum der Schweizerischen Bundesbahn (SBB) in Murthen, das von Prof. Fritz Haller 1987 geplant wurde.

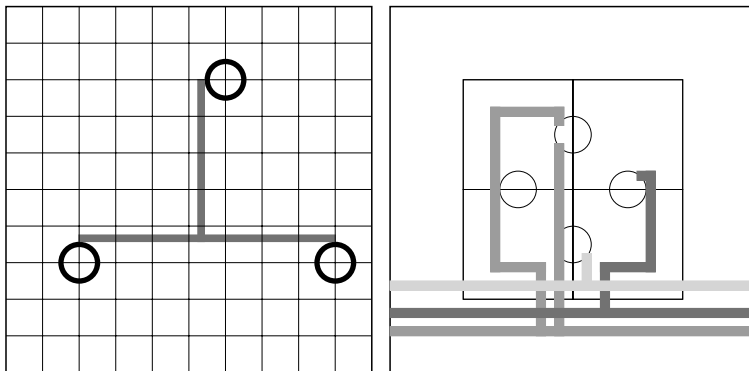
Die vorliegende Arbeit steht in der Tradition der langjährigen Grundlagenforschungen am Institut für Industrielle Bauproduktion unter der Leitung von Prof. Fritz Haller. Er entwickelte in den sechziger Jahren verschiedene Gebäudebaukästen (Bild 1), (Haller 74) (Haller 88), die es ihm erlaubten, seine Architektur sehr schnell an neue Anforderungen anzupassen. 1977 begannen seine Bemühungen an der Universität Karlsruhe, die Planung und Verwaltung dieser Gebäude mit Computern zu unterstützen. Zunächst konzentrierte sich die Forschung auf die schwierigste Teilaufgabe, die technischen Systeme. Dabei stellte sich schnell heraus, daß - besonders für sich schnell ändernde komplexe Gebäude - an Gebäudeentwurf und Gebäudeverwaltung die gleichen Anforderungen zu stellen sind: Es wurde der Begriff permanentes Design für dynamische Gebäude geprägt. Das Institut sieht Computer als integralen Bestandteil eines Gebäudes an. Sie ermöglichen die Verbesserung von Entwurf, Konstruktion und Betrieb.

Erstes Ergebnis dieser Forschung ist das allgemeine Installationsmodell ARMILLA (Anhang II), eine Methodik für die räumliche Koordinierung und den kooperierenden Entwurf der technischen Ausrüstung komplexer Gebäude. ARMILLA konnte 1985 beendet werden (Haller 85) und wird z.Zt. im Architekturbüro von Prof. Fritz Haller, noch ohne besondere Computerunterstützung, eingesetzt.

Am Institut entstanden in der Folge vier Softwareversionen von ARMILLA:

## Objekte und Regeln

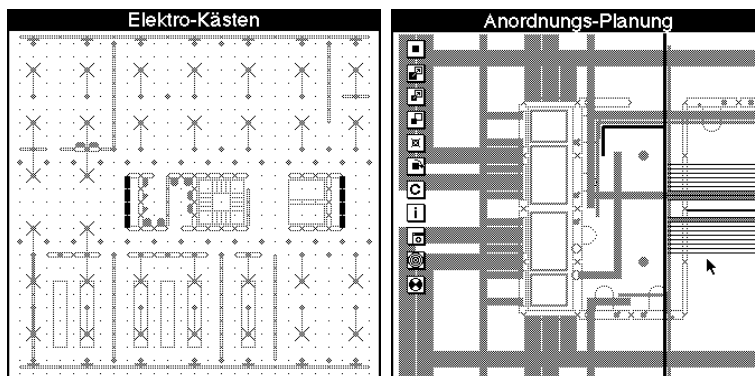
Die erste Version A1 (1986) zeigte, daß eine Implementation von ARMILLA mit Objekten und Regeln möglich ist (Bild 2), (Mathis 88). In der Folge wurde die KI-Shell KnowledgeCraft (KC 90) für mehrere kleine Experimente benutzt. Dabei entstand das Problem, daß selbst kleine Teilaufgaben auch mit sehr umfangreichen Regeln nicht in jedem Fall fehlerfrei bearbeitet werden konnten.



**Bild 2** Zwei typische Bilder des Prototyps A1. Es konnte gezeigt werden, daß die grundsätzlichen Arbeitsschritte von ARMILLA mit Objekten und Regeln beschrieben werden können.

## Blackboard System

Deshalb wurde mit der zweiten Version A2 (1988) die Idee eines "Planungsautomaten" zugunsten von "Planungsassistenten" aufgegeben. Die Forschung konzentrierte sich in der Folge auf den iterativen und mehrstufigen Designprozeß (Bild 3). A2 wurde mit KnowledgeCraft in der Art eines Blackboard-Systems mit einem einfach zu bedienenden, CAD ähnlichen Interface implementiert. Ein besonderer Schwerpunkt von A2 war die Gestaltung eines gut funktionierenden Dialoges zwischen Nutzern und Expertensystemen auf mehreren Abstraktionsebenen (Hovestadt et.al. 89),(Drach et.al. 90), (Hovestadt 90a)

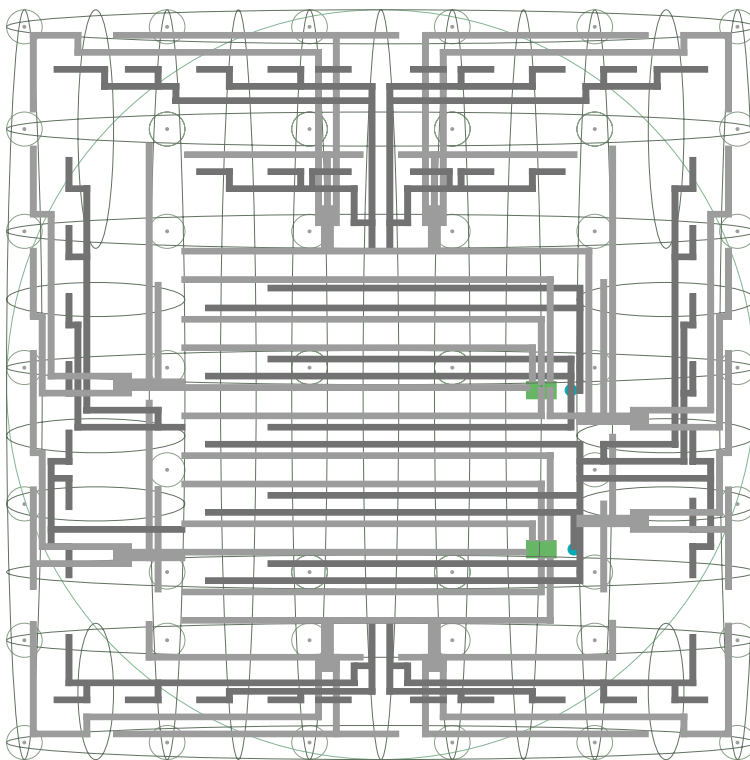


**Bild 3** Zwei typische Bilder des Prototyps A2. Im Vergleich zu A1 können Projekte von größerer Komplexität geplant werden, weil die Programme nicht mehr als Planungsautomat, sondern als Planungsassistenten angesehen wurden.

## Multiuser Blackboard System

Die Version A3 (1990) versuchte die Prinzipien der Vorgängerversion sowohl auf eine realistische Datenmenge als auch auf Mehrnutzerbetrieb zu erweitern. Diese Version hatte keinen Erfolg, weil es nicht möglich war, die entstehende Komplexität (gleichzeitiges, interaktives und mehrstufiges Design in großen und schwach strukturierten Problemfeldern) mit einer zentralen Kontrolle, wie sie bei Blackboard Systemen oder objektorientierten Ansätzen nötig ist, zu verwalten.

## Der vieldimensionale Datenraum



**Bild 4** Ein typisches Bild des Prototyps A4. Der Verzicht auf eine einheitliche Strukturierung der Daten zugunsten eines einheitlichen Interfaces ermöglicht die weitgehende Unterstützung von Planungen relevanter Komplexität.

A4 ist die vierte Softwareversion von ARMILLA (Bild 4). Sie verzichtet auf jede zentrale Kontrolle und verfolgt die These, daß Planer und Nutzer ein Gebäude nicht, wie üblich 3-dimensional, sondern in einem vieldimensionalen Datenraum planen und betreiben können. Ein solches, mit A4 geplantes Gebäude könnte auf eine einheitliche und visuell leicht zu kontrollierende Weise alle Informationen über seinen Lebenszyklus, d.h. Entwurf, Konstruktion und Betrieb, enthalten. A4 ist dabei vom Ansatz her mehrbenutzerfähig, kann in Teilbereichen automatisiert werden und ist dennoch inhaltlich und formal weitgehend offen. Dabei hat sich die neue Version soweit von ARMILLA emanzipiert, daß sie auch für andere Architekturen relevant sein kann.

Die in A4 entwickelten experimentellen, grafischen Darstellungstechniken, Interfaces und Implementationen versetzen mehrere Nutzer in die Lage ohne besondere Anstrengungen (d.h. mit einer Performance und einem Strukturierungs- und Organisationsaufwand herkömmlicher CAD-Systeme) 20- bis 100-tausend Objekte



(d.h. Bauteile, Leistungen, Aufgaben, Funktionen, Räume ...) im vieldimensionalen Datenraum anzuordnen und zu benutzen. Diese Objekte können sowohl den Entwurf, die Konstruktion als auch den Betrieb eines Gebäudes betreffen und - wenn der Aufwand einer Implementation lohnt - teilweise automatisiert bearbeitet werden. Gleichzeitig ist die Struktur von A4 aber so offen, daß v.a. in frühen Planungsphasen oder in Randbereichen leicht Alternativen und neue Strukturen erprobt werden können.

A4 ist bis jetzt noch experimentell und prototypisch anzuwenden: Die vorgeschlagene Strukturierung von Gebäuden im vieldimensionalen Datenraum genügt noch keinen professionellen Anforderungen, die bearbeitbare Datenmenge ist noch um eine Größenordnung zu klein und die für den vieldimensionalen Datenraum entwickelten Darstellungs- und Navigationstechniken sind für einen Nutzer in der Praxis noch zu abstrakt. Dennoch ergeben sich aus dieser Arbeit für die Bereiche Gebäudemodelle, Visualisierung komplexer Zusammenhänge, Künstliche Intelligenz und Datenbanken interessante neue Aussagen und Anforderungen.

## 2 Das Problem

Der **Bauprozeß** (üblicherweise auch Bauen genannt) ist - anders als viele andere Bereiche, in denen Computer eingesetzt werden - schwach strukturiert. Er ist diskursiv, explorativ und benutzt pragmatisch sehr vielfältige Werkzeuge. Gleichzeitig ergeben sich durch neue technische und planerische Möglichkeiten wiederum komplexere Bauaufgaben, die ihrerseits wieder leistungsfähigere Werkzeuge erfordern. Ein Überblick über **Computeranwendungen** im Bauen zeigt, daß es sehr leistungsfähige Spezialwerkzeuge gibt, die aber z.Zt. noch nicht so aufeinander abgestimmt sind, um den versprochenen Nutzen bringen zu können. Am Ende dieses Kapitels werden verschiedene Ansätze für eine **Integration** beschrieben. Sie finden sich nicht nur in Klassifikationen oder Gebäudemodellen, sondern z.B. auch bei der Beschreibung von Arbeitsprozessen oder der Architektur selbst.

## Der Bauprozeß

### one-of-a-kind

Unter Bauprozeß soll hier die Bauplanung, -steuerung und -verwaltung eines Gebäudes während seines Lebenszyklus verstanden werden. Dieser Prozeß erstreckt sich über einen langen Zeitraum, in Zusammenarbeit vieler Personen, die sehr vielfältige Planungs-, Verwaltungs- und Kommunikationsmittel benutzen. Besonders die Planungsarbeiten bewegen sich in verschiedenen Beschreibungsebenen und finden unter verschiedenen Aspekten statt; sie sind explorativ, nicht deterministisch, stehen unter Zeitdruck und müssen daher pragmatisch sein. Ein Gebäude wird individuell geplant und betrieben (one-of-a-kind). Das Bauen ist - anders als viele andere Bereiche in denen Computer eingesetzt werden - eine besonders umfangreiches Einsatzgebiet.

### Die heutige Arbeitsweise

Als Beispiel für die heutige Arbeitsweise soll die Hochbauplanung in der Bundesrepublik Deutschland dienen. Sie läuft in der Regel nach den in den Honorarordnungen (HOAI 91) festgelegten Phasen ab. Man unterscheidet Grundlagenermittlung, Vorplanung, Entwurfsplanung, Genehmigungsplanung, Ausführungsplanung, Vergabe, Bauausführung und Objektbetreuung mit Dokumentation. Jeder Spezialist (Haustechnikingenieur, Bauphysiker, Akustiker, Lichttechniker etc) führt - meist unter der Leitung des Architekten als Treuhänder des Bauherrn und Gesamtkoordinator - zum gegebenen Zeitpunkt die von ihm verlangten Planungsschritte aus, ohne allzusehr auf die Belange der anderen Gewerke Rücksicht zu nehmen. Dieser vor allen Dingen in handwerklichen Traditionen begründeten Arbeitsweise stellen sich neue Anforderungen (s.u.). Das Baugewerbe reagiert mit neuen Arbeitskonzepten, z.B. der sogenannten Integralen Planung (Suter et al. 86): statt die einzelnen Gewerke isoliert zu beplanen und nur deren Ergebnisse aufeinander abzustimmen, wird versucht, durch interdisziplinäre Planungsteams das Gesamtgebäude zu optimieren.

## Ausweitung der Planungsaufgaben

Die Planungsaufgaben werden erheblich erweitert. Bei sinkendem Bauvolumen wächst der Anteil für Umbau, Sanierung und Bauerhaltung zuungunsten des Neubaus. Die jährlichen Baufolgekosten steigen gegenüber den anfänglichen Baukosten sehr stark. Sie liegen in der Regel zwischen 10% und 30% der anfänglichen Baukosten: Verkehrsanlagen 10%, Kläranlagen 20%, Krankenhäuser 26%, Kindergärten 30% (Keller 86). In hochspezialisierten Fertigungsanlagen liegt der Anteil der Baufolgekosten oft schon über 100%.

Diese Strukturänderungen im Bauwesen ziehen Änderungen im Leistungsbild der Planung nach sich: Der Schwerpunkt verschiebt sich eindeutig von der Neuplanung hin zur Nutzungsplanung, Instandhaltungsplanung, Instandsetzung und wirtschaftlichen Verwertung von Gebäuden. Dafür haben sich die Begriffe Objekt-Management, Projekt-Management oder Facility-Management eingebürgert (Richter 88 p 9), (Worthington 89). Gebäude, deren Design, Konstruktion und Management zunehmend integriert werden, werden Automated Buildings genannt (Vincent et al. 85).

## Intelligente Gebäude

Gebäude werden in zunehmendem Maße mit Informationstechnik ausgestattet. Ein gutes Beispiel für diesen Trend ist die aktuelle Diskussion um "Intelligente Fassaden". Sie reagieren automatisch auf die äußeren klimatischen Bedingungen mit dem Ziel möglichst konstanter, ökonomischer, ökologischer und subjektiv angenehmer innerer klimatischer Bedingungen. Diese Fassaden werden sogar als Kommunikations- und Gestaltungsmittel entdeckt (Müller H. 89), (Gauchel 89), (Friedrichs 91), (ARCH+ 90), (ARCH+ 91), (Kohler 91). In Tokyo wurde 1989 ein Demonstrationsgebäude erstellt, dessen Steuerungseinrichtung ca. 1000 Mikroprozessoren enthalten (Tron 89).

## Spezialisierung und Diversifizierung

Das Bauen diversifiziert sich unter verschiedenen Aspekten:

- Die steigende Komplexität der Bauteile geht einher mit einer steigenden Menge von Bauteilen verschiedenster Art, die in einem Gebäude verwendet werden.
- Die Bauaufgaben spezialisieren sich sehr stark. Konnte im 18/19.Jh. noch eine Gebäudetypologie in zwei Bänden zusammengestellt werden (Durand 1801), (Durand 1802), so wird es schwer sein, heute überhaupt einen Katalog der Bauaufgaben aufzustellen. Kindergärten, Kantinen, Klärwerke, Kraftwerke etc. sind Bauaufgaben mit neue Anforderungen.
- Der Umfang von Normen, Standards und Prüfverfahren steigt kontinuierlich mit der Zahl der Bauaufgaben, Baumaterialien und Bautechniken. Ein gutes Beispiel sind die gestiegenen Anforderungen an einen möglichst geringen Energieverbrauch, die nur mit erheblichem technischem, strukturellem, planerischem und organisatorischem Aufwand zu erreichen sind (Payne 89).
- In direktem Zusammenhang damit steht die Zunahme der am Bauprozeß beteiligten Personen. Gab es noch vor 150 Jahren einen Baumeister, der tatsächlich den gesamten Bauprozeß koordinierend überblicken konnte, so spalteten sich zunächst die Bauingenieure und mit der Zeit immer mehr Spezialisten ab, so daß heute nicht selten 20 Ingenieure verschiedener Fachrichtungen am Bauprozeß beteiligt sind und nur schwer eine gemeinsame Planung formulieren können.

Im Interesse einer Dämpfung v.a. der Betriebs- und Wartungskosten der Gebäude ist jedoch eine stärkere Standardisierung und Typisierung erwünscht.

## Internationalisierung

Die Diversifizierung und Spezialisierung geht einher mit der Vergrößerung der Märkte. Spezialleistungen werden schon jetzt weltweit angeboten. Es werden Probleme dadurch entstehen, daß Planungen international entwickelt werden können, die Ausführung aber in weiten Bereichen auf regionale Strukturen angewiesen ist.

## Dynamisierung

Mit der Diversifizierung und Spezialisierung der Baukomponenten steigt auch die Anzahl der Innovationszyklen. Während Anfang des Jahrhunderts von 30 - 50 Jahren ausgegangen werden konnte, müssen heute wesentliche Komponenten eines Gebäudes nach 5 - 10 Jahren ausgetauscht werden (Banham 60, 69, 81), (Worthington 89), (Pawley 90), (ORBIT1 83), (ORBIT2 85). Damit die Gebäude nicht nach so kurzer Zeit nutzlos werden, sollte man vermehrt darauf achten die Gebäude so zu modularisieren, daß die schnell alternenden Komponenten reibungslos ausgetauscht werden können.

## Informatisierung

Die Dynamisierung und Spezialisierung des Bauens steht in Zusammenhang mit Entwicklungen, die als das "Zweite Maschinenzeitalter" bezeichnet werden (Banham 60), (Pawley 90): Hochentwickelte Massenprodukte (Auto, TV ...) ermöglichen einen permanenten Gebrauch und haben dadurch, anders als die Maschinen des "Ersten Maschinenzeitalters" (Eisenbahn, Kino ...), substantiellen Einfluß auf das tägliche Leben. Gleichzeitig führt die Entwicklung der Informationstechniken dazu, daß Planerteams weltweit zusammenarbeiten werden ("global village") (McLuhan 64), (Powers 80), (Flusser 89).

Diese Entwicklung wiederum hat grundlegenden Einfluß auf die Gebäudeorganisation und -struktur. Nach Worthington 91 gibt es einen Trend zu kleinen, intelligenten Einheiten, in denen Kleingruppen sehr gut ausgebildeter Angestellter hochgradig miteinander vernetzt und eigenverantwortlich sehr vielfältige Arbeiten verrichten (ORBIT1 83), (ORBIT2 85), (Duffy et al. 89). Die Gebäudeeinheiten werden entsprechend kleiner, die Nutzungen vielfältiger: an die Stelle der großen Bürotürme oder Fabrikanlagen treten kleinere und sehr vielfältig genutzte Einheiten. Es werden neue Gebäudetypen für den wachsenden Bedarf an Aus- und Weiterbildung entstehen, als deren Prototyp das Centre Pompidou gelten kann. Es werden Infrastrukturen für die wachsenden Verfügungs- und Bewegungsbereiche der kontinuierlich wechselnden Arbeitsgruppen geschaffen werden (Haller 75).

# Computeranwendungen

*Betrachtet man die Software, die zur Zeit im Bauwesen eingesetzt wird, so findet man verschiedene Entwicklungslinien, die jeweils nur einen Aspekt betreffen. Die Software ist zwar leistungsfähig, aber in der Regel auf ein Spezialgebiet beschränkt. Wegen dieser mangelnden Integration sind die Einsatz- und Entwicklungsmöglichkeiten auf ein Spezialgebiet beschränkt.*

*In diesem Kapitel werden acht verschiedene Entwicklungslinien anhand einzelner Beispiele beschrieben. Diese Beispiele sind zufällig ausgewählt und dienen nur zur Illustration des momentanen Entwicklungsstandes. Für eine repräsentative und ausführliche Übersicht gibt es einschlägige Kataloge und Ausstellungen (ISIS Engineering Report 91), (IAP 91), (ACS 91).*

## Spezialisierung

Die allgemein bekannten Programme für Text, DeskTop Publishing (DTP), Grafik, Computer Aided Design (CAD), Tabellenkalkulation etc. entwickeln sich von den unspezifischen Allround-Werkzeugen hin zu sehr leistungsfähigen Spezialprogrammen: RADIANCE kann auf der Grundlage von CAD-Daten Beleuchtungen, Wärmestrahlung oder akustische Probleme teils sogar photorealistisch darstellen. BLAST oder DOE2 ermitteln ebenfalls auf der Grundlage von CAD-Daten die Energiebilanzen eines Gebäudes abhängig von den inneren und äußeren Bedingungen über einen bestimmten Zeitraum. CLIMESTRA unterstützt sehr wesentlich den Entwurf von Regelungssystemen für Klimaanlage, ermöglicht Simulationen und ermittelt den Energiebedarf. ALLPLAN ermöglicht auf der Basis von CAD-Techniken weitgehend computer-unterstützte Berechnung und Layout von Bewehrungen von Stahlbetonteilen. ABACUS oder SOFiSTiC, als Repräsentanten von FE (Finite Elemente) Programmen, ermöglichen die statische und dynamische Berechnung beliebiger Konstruktionen. GIP ermöglicht das schnelle Erstellen von Ausschreibungstexten anhand von vordefinierten Standardtexten in Kombination mit Tabellenkalkulation. ISICAD oder VISIONAEL bieten auf der Basis von Datenbanksystemen eine ganze Palette von einfach zu bedienenden integrierten Programmen besonders für die Gebäudeverwaltung, die von mehreren Personen gleichzeitig bedient werden können.

## Künstliche Intelligenz

Der Gebrauch von Algorithmen für deterministische Teilprobleme wird ergänzt durch Methoden der Künstlichen Intelligenz für nicht-deterministische Teilprobleme. Auf der kommerziellen Ebene ist es möglich, kundenspezifische Module auf der Grundlage eines allgemeinen Systemkerns mit Hilfe von KI-Techniken sehr schnell zu entwickeln (z.B. MTEL-IA). Mit Hilfe von Entscheidungstabellen kann ein erster, von den Nutzern selbst vorgenommener, Schritt in Richtung KI gemacht werden (z.B. VISIONAEL). Für einen weitergehenden Einsatz von KI-Techniken bietet sich NEXPERT-OBJECT an, für das nur wenige Programmierkenntnisse erforderlich sind. Leistungsfähiger sind BABYLON, KEE oder KNOWLEDGE-CRAFT. Sie sind echte Entwicklungsumgebungen für KI-Systeme und setzen entsprechendes Wissen voraus. Weil diese Shells auf LISP basieren, sind sie Exoten. Sie benötigen große Rechnerkapazitäten und können dennoch nicht die erforderlichen Datenmengen in akzeptabler Zeit bearbeiten. Neuere Shells basieren auf C und C++ und könnten eine breitere Akzeptanz finden (z.B. CLIPS).

## Datenmenge

Nicht nur die Komplexität, sondern auch die reine Datenmenge übersteigt die Leistungsfähigkeit heutiger Softwaretechniken: so gibt es z.B. im Flughafen München ca. 100.000 Sensoren, die asynchron und z.T. mit Echtzeitanforderungen abgefragt und bearbeitet werden müssen. Gleichzeitig müssen kontinuierlich Änderungen über Jahrzehnte möglich sein ohne den laufenden Betrieb zu unterbrechen. Ein solches Gebäude besteht darüber hinaus aus Millionen von Baukomponenten, die von hunderten von Planern spezifiziert wurden, die von tausenden von Personen montiert und von hunderten von Personen über die nächsten Jahrzehnte gewartet werden müssen. Große Datenbanksysteme wie z.B. ORACLE oder INGRES können prinzipiell Datenmengen dieser Größenordnung verarbeiten, erkaufen sich diese spezifische Leistungsfähigkeit aber mit sehr restriktiven Verwaltungsmechanismen, die sich dem enormen Veränderungsdruck dieser Gebäude widersetzen. Objektorientierte DBMS wie z.B. ONTOS oder ITASKA bieten zwar mehr Flexibilität und können auch eine größere Vielfalt von Daten verwalten, konnten aber noch nicht zeigen, daß sie die geforderten Datenmengen bewältigen können.

## Nutzerkontrolle

Wichtig ist auch die direkte und möglichst schnelle interaktive Kontrolle eines Systems. Die mittlerweile allgemein bekannten Möglichkeiten grafischer dreidimensionaler Datenrepräsentation bis hin zur sogenannten Virtual Reality zeigen diese Entwicklungslinie. Eine wichtige Rolle spielen hier Künstler aller Art (ARS ELEKTRONICA) oder auch Videospiele. In Bezug auf Architektur sind die Arbeiten (Negroponte 70, 75) und (MediaLab) wichtig. Weniger spektakulär, aber vermutlich tragfähiger, sind die Forschungsarbeiten an leistungsfähigeren 3D Interfaces am Paolo Aalto Research Center (Clarkson 91).

## Vernetzung

Eine wesentliche Entwicklungslinie ist durch die globale Vernetzung der Computer gekennzeichnet (McLuhan 64), (Powers 80). Sie soll dazu führen, daß alle am Bauprozeß beteiligten Personen trotz räumlicher Trennung "wie an einem Plan" zusammenarbeiten können.

## Multimedia

Unmittelbar vor der breiten Markteinführung stehen die sogenannten Personal Workstations, die auf der Basis von UNIX zunächst eine Palette von Basiswerkzeugen für Text, Ton, Bild, Grafik, Tabelle ... anbieten, die untereinander gekoppelt die intensive Zusammenarbeit vieler Personen ermöglichen. Wegbereiter dieser Entwicklung ist das Paolo Aalto Research Center. Produkte gibt es z.Zt. von NEXT und SILICON GRAPHICS. Eine berühmte Vision dieser Entwicklung formulierte Apple mit dem KnowledgeNavigator.

## Programmierung durch Anwender

Als letzte wichtige Entwicklung können Programme angesehen werden, die sehr einfach zu bedienen sind und daher von ihren Nutzern geändert und angepaßt werden können. Es gibt hier einen fließenden Übergang zur Ausbildungssoftware. Als Beispiele können gelten: MATHEMATICA für symbolische Mathematik, HYPERCARD für die flexible Verwaltung kleiner multimedialer Datenmengen, NEXT mit seinem INTER-FACE-BUILDER und APPLICATION-KIT als objektorientierte Arbeitsumgebung und/oder Programmierumgebung, mit Abstrichen das Prinzip der Tabellenkalkulation z.B. EXCEL oder die Erweiterungsmöglichkeiten von AUTOCAD durch die Programmiersprache AUTOLISP.

# Integrierende Strukturen

*Grundlage für einen erfolgreichen Informationsaustausch sind neben allgemein anerkannten Inhalten Übereinkünfte in deren Strukturen. Bemühungen um solche "Sprachen" finden sich mit verschiedenen Schwerpunkten für verschiedene Bereiche des Bauens. In diesem Kapitel werden Klassifikationen der für ein Gebäude relevanten Informationen, Gebäudemodelle, Bemühungen um eine Formalisierung des Entwurfsprozesses, Gebäudebaukästen und Gebäudetypologien beschrieben. Sie unterscheiden sich in ihrer Kapazität (wieviele Informationen können integriert werden?), Leistungsfähigkeit (was kann mit den integrierten Informationen gemacht werden?) und Offenheit (wie vielfältig können die Informationen sein, können Informationen mit neuen Strukturen aufgenommen werden?).*

## Klassifikationen

Als gutes Beispiel für eine integrierende Struktur im Bauwesen, die durch große Kapazität und Offenheit aber geringer Leistungsfähigkeit charakterisiert ist, kann die SfB (Samarbetskommitten för Bygguardsfragor = Koordinierungsgruppe für die Bauwirtschaft) dienen (Bindslev 69), (Piel 78). Sie ist ursprünglich eine schwedische Entwicklung, die auf die Jahre 1948/49 zurückgeht. Das SfB-System gehört zur Gruppe der Facetten-Klassifikationen und wird seit 1950 kontinuierlich angewandt, um Normen, Standardleistungsverzeichnisse, Preislisten und Daten für Produktinformationen zu organisieren. Seit Anfang der sechziger Jahre findet das System in zunehmendem Maße internationale Anerkennung und Verwendung (besonders in Skandinavien, England, der Tschechoslowakei, Japan, später auch in den Benelux Ländern, der Schweiz und Deutschland). Schnell gab es internationale und nationale Koordinierungsgruppen und das ständige Bemühen, das SfB-System zu erweitern, zu verfeinern und an die nationalen Besonderheiten anzupassen. SfB ist so verbreitet, daß manche Fachzeitschriften ihre Artikel und sogar Firmen ihre Prospekte mit dem SfB-Code versehen.

Der leichte Zugang für einen Anfänger zum System, seine Nutzerfreundlichkeit, seine Integration in die herkömmlichen Arbeitsverfahren und seine Offenheit für Entwicklungen, Erweiterungen, Verfeinerungen und nicht strukturierte Informationen sind Voraussetzungen für die Akzeptanz einer Klassifikation. Diese Anforderungen erfüllen die neueren, auf DV-Systemen basierenden Ansätze weniger als das SfB-System.

Der besonderen Nutzerfreundlichkeit und Offenheit der SfB-Systeme auf der technischen Ebene stehen starke Restriktionen auf der logischen Ebene gegenüber: Das SfB-System ist durch die Methodik und Logik von Karteikarten bestimmt und ist daher ein Instrument für die Informationsverwaltung, das keine Ansätze für eine Verarbeitung der Informationen bietet. Deswegen kann es nur sehr mittelbar der Kommunikation und dem kooperierenden Planen dienen.

## Relationale Datenbanken zur Gebäudeverwaltung

Eine Weiterentwicklung auf der logischen wie auf der technischen Ebene ist z.B. der Ansatz von Richter (Richter 88) mit seiner "Integrierten Informationsstruktur für Relationale Datenbanken im Bauwesen". Richter übernimmt die verschiedensten üblichen Verfahren, Normen, Klassifikationen etc. und konstruiert damit ein logisches System, das potentiell alle Informationen des Bauprozesses umfassen kann. Diese formale semantische Beschreibung des Bauprozesses ist bemerkenswert umfassend, detailliert, kompetent und offen. Auf der technischen Ebene wurden die Tabellen relationaler Datenbanken benutzt. Im Gegensatz zum SfB können daher die Informationsstrukturen von verschiedenen Nutzern an verschiedenen Orten benutzt und auch durch Automaten bearbeitet werden, sodaß z.B. jede Information nur einmal eingegeben, aber unter

verschiedenen Aspekten genutzt oder weiterverarbeitet werden kann. Richters System ist daher nicht nur für die Informationsverwaltung einsetzbar, sondern es dient potentiell auch der Informationsverarbeitung, d.h. Kommunikation und Kooperation können unmittelbar unterstützt werden. Das Problem ist jedoch, daß die eingesetzte Technik relationaler Datenbanken so restriktiv ist, daß ein Nutzer seine hergebrachten Werkzeuge und Darstellungstechniken meistens nicht weiter nutzen kann, und streng den vordefinierten Informationsflüssen folgen muß. Änderungen, Erweiterungen oder Verfeinerungen der Arbeit können nur durch einen Informatiker geschehen und liegen damit jenseits der Möglichkeiten eines Nutzers.

Das logische und technische Gesamtsystem von Richter eignet sich daher nur für Planungs- und Verwaltungsaufgaben, die sehr übersichtlich und invariant sind und an denen über eine lange Zeit viele Personen immer wieder die gleichen Arbeiten verrichten müssen. Es ist daher weder für einen Gebäudeentwurf geeignet (dazu fehlen dem System v.a. Strukturen für verschiedene Abstraktionsebenen und unscharfe Informationen), noch ist es für die neuen, sehr komplexen und dynamischen Gebäude oder für die Arbeit kleiner Planungsbüros geeignet, die sich schnell auf kleine und sehr unterschiedliche Planungsaufgaben einstellen müssen und diese mit nur wenigen Personen bearbeiten. Es ist aber sehr wohl für sehr große Projekte und große Planungsbüros oder Behörden geeignet.

Bei einer Implementation des logischen Modells mit moderneren technischen Mitteln sind jedoch mehr Offenheit und Flexibilität zu erwarten.

## **Gebäudemodelle**

Auf einer eher experimentellen Ebene stehen die Bemühungen um Gebäudemodelle. Bei ihnen werden in der Praxis etablierte Standardaustauschformate oder Datenbanken benutzt. Um eine möglichst große Leistungsfähigkeit zu erreichen, geben sie aber eine für die Praxis ausreichenden Kapazität und Offenheit auf.

Gebäudemodelle oder allgemein Produktmodelle haben ihren Ursprung in den Datenaustauschformaten von CAD-Systemen. Diese Entwicklung ist z.Zt. vom Maschinenbau dominiert. Es sind jedoch entsprechende Entwicklungen auch in der Bauindustrie vorstellbar. Während die ersten CAD-Systeme reine Grafik Systeme waren und daher nur grafische Daten austauschten (z.B. durch IGES), gibt es heute einen Trend zu Produktmodellierern (Anderl 91), bei denen alle Informationen, die ein Produkt betreffen, von der Produktfindung über die Produktrealisierung und Markteinführung bis zur Marktsättigung und Produktionsauflösung integriert verarbeitet werden sollen. Für den Maschinenbau werden neben einem Geometriemodell auch Toleranzmodelle, Oberflächenangaben, Materialeigenschaften, Technologien, technische Gestaltung, Funktionsmodelle, Fertigungsplanungsmodelle, Montageplanungsmodelle, Meß- und Prüfmodelle, Produktionsplanungsmodelle und Darstellungsmodelle benötigt. Die Entwicklungsphasen der Modelle werden mit Konzeptionierung, Spezifikation, Validierung und Implementation bezeichnet. Diese Modelle sind noch weitgehend in der Diskussion. Es gibt jedoch vereinzelte Projekte, die sich mit einer Integration befassen. Eine gute Übersicht geben die internationalen Bemühungen um standartisierte Datenaustauschformate für Produktmodelle, z.B. STEP (Wike 89), (Anderl 87). Für den Baubereich wurde im Zusammenhang mit STEP ein besonderes Anwendungsmodell entwickelt (STEP-2DBS 89). Es ist allerdings z.Zt. nur in der Lage (fehlerhaft!) 2d Grafik-Informationen auszutauschen. Die Entwicklungen im Maschinenbau sind aber dennoch interessant für das Bauwesen, weil sie mit großer Energie betrieben werden, weil sie sich verstärkt auf sehr flexible Produktion kleiner Stückzahlen konzentrieren, und weil sie vor allen Dingen durch die zunehmende Arbeitsteilung (z.B. verteilt die Automobil-Industrie zunehmend die Produktion auf ihre Zulieferer) immer mehr Anforderungen an ihre Arbeitsorganisation stellen, die denen der Bauindustrie verwandt sind.

Dennoch ist es ein Fehler, die Methoden des Computereinsatzes der multinationalen Konzerne z.B. im Maschinenbau einfach auf die kleinen bis mittelgroßen Betriebe im Baugewerbe zu übertragen, wie z.B. in (CIB 91), (Augenbroe 89: COMBINE), (Enkovaara 88: RATAS). In der Bauindustrie gibt es keine zentrale



Instanz, die einen top-down Ansatz entwickeln, durchsetzen und nutzen könnte. Hier müßten sich die Werkzeuge, Normen und Standards vielmehr bottom-up und allmählich am unmittelbaren Kommunikationsbedarf entwickeln.

Auch sind die bisherigen Bemühungen um Produktmodelle stark von dem gerichteten CAD-CAM Datenaustausch geprägt. Für die ungerichtete Zusammenarbeit verschiedener Planer wird ihre Eignung überschätzt.

## Formalisierungen des Entwerfens

Die bisher beschriebenen Strukturen verfolgten das Ziel, möglichst viele Informationen auf der sicheren Grundlage eines fertigen Gebäudes zu integrieren. Die unsicheren Informationen während des Gebäudeentwurfes sind außerhalb ihrer Möglichkeiten. Eine Fülle von Integrationsansätzen konzentrieren sich auf den Gebäudeentwurf, erhöhen dafür die Leistungsfähigkeit, müssen gleichzeitig aber weite Bereiche des Bauprozesses ausklammern. Forschungen über eine Formalisierung des Entwerfens benutzen entweder Datenbanken oder Methoden der Künstlichen Intelligenz. Erstere behandeln im Vergleich viele Daten geringerer Leistungsfähigkeit mit professionellen Werkzeugen, letztere wenige Daten hoher Leistungsfähigkeit mit experimentellen Werkzeugen.

Ein gutes Beispiel für eine Formalisierung im Bereich des Entwerfens mit Hilfe von Datenbanktechniken ist das GARM (General AEC Reference-Model) Modell (Gielling 88), (Waard 91). Es werden drei verschiedene Strukturen unterschieden:

- Ein räumliches System mit der Relation 'consist-of
- Ein Baukomponenten System mit den Relationen 'formed-by bzw. 'decomposed-in. Eine Kopplung mit dem räumlichen System geschieht über die sehr interessante Relation 'space-boundary.
- Darüber hinaus wird in ein sogenanntes Functional Unit (FU) für die Beschreibung der Anforderungen und eine sogenannte Technical Solution (TS) unterschieden. Beide sind durch die Relation 'fulfills verknüpft.

Diese drei Strukturen können hierarchisiert werden und sind im Ergebnis eine neuartige und gut verständliche Abbildung der räumlich-funktionalen Sachverhalte eines Gebäudes.

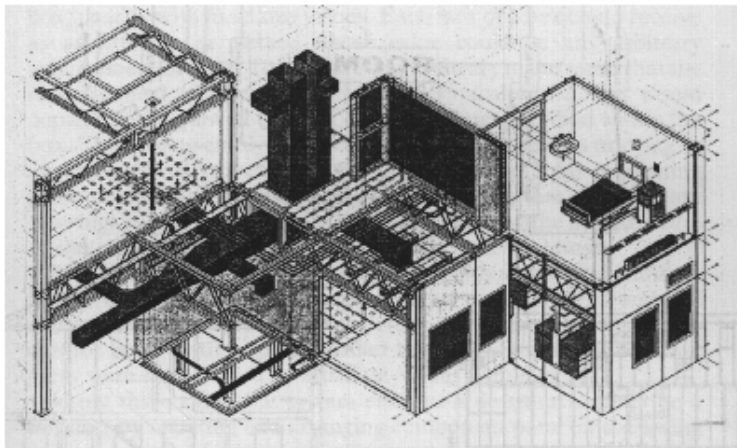
Obwohl das Modell darauf abzielt, eine weitergehende Computerunterstützung v.a. für die funktionale und räumliche Koordination von Gebäuden zu ermöglichen, gibt es noch keine Beispiele, in denen diese Möglichkeiten ausgenutzt wurden. Auch gibt es außer der graphisch unterstützten Beschreibungssprache NIAM (Nijssen 89) mit der über dem ISO Standard für Produkt-Modellierung (EXPRESS 89) eine relationale Datenbank spezifiziert wird, noch keine Interfaces für die entstehenden, sehr komplexen hierarchischen Datenstrukturen. Deshalb ist auch noch nicht gezeigt, wie leistungsfähig dieses System von de Waard tatsächlich sein kann.

In der KI Diskussion gibt es ein traditionelles Interesse an Formalisierungen des Designprozesses. Hier wird Design als ein Suchprozeß nach einer geeigneten Lösung aus der Menge aller Lösungen (Designraum) beschreiben. Im Designraum sind die Designprobleme in Teilprobleme zu Hierarchien gegliedert. Ein Designschritt besteht in der Regel aus dem Erstellen eines (Teil-)Planungszieles (design formulation), dem Erzeugen von einer oder mehreren (Teil-)Lösungen (design synthesis) und abschließend der Auswahl der geeignetsten (Teil-)Lösung (design evaluation). Es wird unterschieden zwischen dem Wissen, ein neues (Teil-) Planungsziel zu erstellen (planning knowledge) und dem Wissen, eine geeignete (Teil-) Lösung zu entwickeln (design knowledge). Benutzt werden viele Kombinationen der Wissensrepräsentationstechniken, wie z.B. Frames, Regeln, Constraints oder Truth Maintenance. Beispiele sind (Mitchel 85: VEDEX), (Tong 87: DONTE), (Maher 87b: EDESYS), (Takala 87), (Tomiyaama 87b), (Flemming et al. 88: LOOS), (Coyne

91:ABLOOS). Ältere, nicht mehr aktuelle formale Planungsmethoden in der Architektur finden sich z.B. in (Luckmann 70:AIDA), (Eastman 68), (Asimov 62), (Ackoff 62), (Jones 69), (Markus 69) oder (Manning 68).

In jüngster Zeit entwickelt sich in der KI Diskussion Kritik an der These, Design als Suche in einem Lösungsraum zu beschreiben, die auf (Newell et al. 72) zurückgeht. Kritisiert wird, daß der Lösungsraum vor jeder Problemlösung vollständig beschrieben sein muß, was dem schwach strukturierten Charakter des Designs (Simon 73) nicht entspricht. Design wird von den Kritikern als explorativ beschrieben: "The design process can thus be thought of as a kind of integrated puzzle-making and puzzle-solving, not just as puzzle solving" (Smithers 92, p 2).

## Baukästen



**Bild 5** Das System OXSYS basiert auf der Addition industriell hergestellter Komponenten in einem orthogonalen Raum (Bijl 89 p 85).

Einen anderen Schwerpunkt kann die Integration der Informationen bekommen, wenn es sich nicht um beliebige Gebäude handelt, sondern um Baukästen. Durch die klare Strukturierung der Gebäude und des Bauprozesses bietet sich eine weitgehende Computerunterstützung an. Hier sollen das OXSYS und das MIDI/ARMILLA System vorgestellt werden.

Das OXSYS-System (Bild 5) wurde in den 60er Jahren entwickelt und konnte in den 70er Jahren zur Planung vieler Krankenhäuser von der Oxford Region Health Authority benutzt werden (Hoskins 77) (Bijl 89). Es gliedert Gebäude in diskrete Bauteile, die im Computer als orthogonale Kuben repräsentiert werden können. Diese Kuben enthalten sowohl die graphischen Informationen der verschiedenen Ansichten, als auch Attribute über Materialien, Bestellnummern oder Preise. Das OXSYS erlaubt es einem Planer, durch einfaches Auswählen von Komponenten aus einer Bibliothek und durch das Anordnen dieser Komponenten in einem Plan, ein Gebäude zu beschreiben. Mit Hilfe angegliederter Computerprogramme können dann verschiedene Ansichten gezeichnet, Mengenlisten erzeugt oder Simulationen des Energiebedarfs durchgeführt werden.

Mit dem OXSYS konnte die große Komplexität von Krankenhausplanungen mit den Informationstechnologien der 70er Jahre bewältigt werden. Die Grenzen dieses Systems liegen in seiner prinzipiellen Orthogonalität, der ausschließlichen Verwendung vordefinierter Komponenten und dem additiven Designmodell.

Das Bausystem MIDI/ARMILLA ist Ausgangspunkt dieser Dissertation und wurde in der Einleitung beschrieben. Anders als das oben beschriebene OXSYS-System, das auf der Addition der Baukomponenten

beruht, basiert MIDI/ARMILLA auf einer räumlichen Strukturierung und Koordinierung auf verschiedenen Abstraktionsstufen. Dadurch sind Skizzen ebenso wie Baukomponenten Teil des Systems. So können die verschiedenen Teilsysteme eines Gebäudes schon in frühen Planungsphasen aufeinander abgestimmt werden. Besonders ARMILLA ist deshalb ein originäres Planungsinstrument und in seinem Ursprung kein Verwaltungsinstrument (Anhang II).

## Gebäudetypologien

Einen wesentlichen und für diese Übersicht abschließenden Beitrag zur Diskussion um Strukturierungen in der Architektur findet man in der architekturtheoretischen Diskussion um Typologien, die von den Architekten Krier, Rossi, Botta, Garay, Grassi, Gregotti, Portzamparc, Scolari, Sterling, Ungers und Van Eyck repräsentiert wird und die sich ihrerseits auf Cerda, Saarinen und Otto Wagner bezieht. Sie zeigt, daß Typologien und Strukturierungen der Architektur, wenn sie in Verbindung mit dem Computer stehen (s.o.), sehr stark von den eingeschränkten Möglichkeiten der Datenverarbeitung geprägt sind. Sie bietet gleichzeitig eine tragfähige Begriffsbestimmung von Standards, Typen, technischem und intuitivem Planen etc. (Panerai 80), (Kuhnert 80), (Malfroy 86), (ARCH+ 86), (Muratori 59).

## Anforderungen an ein Integrationsmodell

Für ein neues Modell zur Integration der Informationen eines Gebäudes während seiner Lebenszeit ergeben sich aus dieser kurzen Übersicht folgende Anforderungen:

- Es gibt auf der einen Seite Erfahrungen mit professionellen (in der Praxis eingesetzten) Strukturen, die auf großer Kapazität und geringer Leistungsfähigkeit beruhen. Auf der anderen Seite gibt es Erfahrungen mit ganz anders gearteten experimentellen Strukturen, die auf eine möglichst große Leistungsfähigkeit bei kleineren Datenmengen abzielen. Da ein System mit großer Leistungsfähigkeit und großer Datenmenge nicht realisierbar ist, sollte man versuchen die Ansätze so zu koppeln, daß auf der Basis großer Kapazität und geringer Leistungsfähigkeit in Teilbereichen leistungsfähigere Strukturen aufgebaut werden können.
- Ein vergleichbarer Zusammenhang besteht auch bei der Offenheit der Strukturen: Je leistungsfähiger eine Struktur wird, desto aufwendiger ist es, neue und unvorhergesehene Daten einzufügen. Für den Gebäudeentwurf sind z.B. offene Strukturen für Experimente erforderlich, dagegen benötigt man z.B. für die Gebäudeverwaltung sehr leistungsfähige und etablierte Strukturen. Ein System mit gleichzeitig großer Leistungsfähigkeit und Offenheit ist nicht realisierbar. Man sollte versuchen, beide Anforderungen so zu koppeln, daß auf der Basis offener und wenig leistungsfähiger Strukturen bei Bedarf bereichsweise leistungsfähigere Strukturen installiert werden können.
- Ein wesentliches Problem der bisherigen Integrationsmodelle ist, daß die Strukturen mit zunehmender Leistungsfähigkeit immer schwerer zu kontrollieren sind. Zum Beispiel werden Informationen, die ohne Computereinsatz auf Plänen grafisch verzeichnet wurden, mit Computerunterstützung oft nur noch alphanumerisch verarbeitet. Da dies in den meisten Fällen nicht die adäquate Darstellungsform ist, verliert ein Nutzer schnell den Überblick und die Kontrolle über das System. In einem guten Integrationsmodell sollten also mit den leistungsfähigeren Strukturen auch leistungsfähigere Darstellungs- und Interaktionsmittel entwickelt werden.

# 3 *Das A4-Modell*

## Einleitung

Die weitgehende Computerunterstützung von Entwurf, Konstruktion und Betrieb von Gebäuden ist ein sehr komplexes Problem. Es gibt Millionen von Informationen, organisiert in dutzenden von Dekompositionen (funktionale, kompositorische, räumliche, finanzielle, zeitliche ...). Es sind bis zu tausend Personen beteiligt, die an vielen verschiedenen Orten arbeiten. Sie alle arbeiten asynchron, mit langen, komplexen und sich sehr stark überlagernden Transaktionen (Katz 85). Ihre Arbeit ist explorativ (Müller 89) (Smithers 92). Deshalb benötigen sie vage Daten in vielen Versionen. Gebäude können mit bis zu 100.000 Sensoren und Aktoren (Beispiel: Flughafen München) zur Gebäudesteuerung ausgerüstet sein. Viele dieser Aktoren und Sensoren stellen hohe Anforderungen an Sicherheit und Echtzeit. Die Daten ändern sich kontinuierlich während der Lebenszeit eines Gebäudes (oft mehr als 100 Jahre), weil Gebäude ständig neu genutzt und umgebaut werden. Oft müssen unvorhergesehene neue Techniken in ein Gebäude integriert werden. Entsprechend müssen auch unvorhergesehene Daten in die Datenstrukturen eingebunden werden.

Es gibt eine große Menge leistungsfähiger, aber spezialisierter und voneinander isolierter Software für die verschiedensten Bereiche des Bauens. Es gibt verschiedene Bemühungen um Integrierte Gebäudemodelle, d.h. um eine gemeinsame Datenplattform für die verschiedenen Applikationen. Sie basieren, je nachdem ob es Projekte aus der Praxis oder der Forschung sind, auf relationalen Datenbanken, auf objektorientierten Datenbanken oder Blackboard Systemen. Sie haben alle das Problem, Informationen in großer Menge und großer Komplexität behandeln zu müssen. Gleichzeitig hat ein Nutzer mit den bekannten Browsern und Editoren keine wirkungsvolle interaktive Kontrolle über die komplexen Strukturen. Dies ist ein Grund, warum sich herkömmliche Integrierte Gebäudemodelle auf das Facility Management, Building Performance oder Kostenkontrolle konzentrieren. Sie bewegen sich damit auf der sicheren Plattform existierender Gebäude. Für das Gebäudedesign sind aber flexiblere Strukturen und Werkzeuge erforderlich.

Herkömmliche CAD-Systeme können einen Hinweis geben, wie dieses Problem der flexiblen und interaktiven Kontrolle komplexer Strukturen möglich ist. Sie sind genau in dem Bereich stark, in dem die Integrierten Gebäudemodelle schwach sind: Sie sind in der Lage große Mengen von Daten in einer für einen Nutzer verständlichen und dem Problem angemessenen Form darzustellen und manipulierbar zu machen. Darüber hinaus können sie sehr gut mit geometrischen Informationen umgehen, was für das Bauen von besonderer Bedeutung ist. Auf der anderen Seite sind CAD-Systeme schwach in der Verarbeitung nicht-geometrischer Daten, die die Stärke der Integrierten Gebäudemodelle ist.

Die Hypothese des A4-Modells ist, daß Architekten, Ingenieure und Nutzer ein Gebäude nicht nur, wie üblich, in 3 Dimensionen planen und betreiben, sondern diese Arbeit - wie später ausführlich beschrieben wird - auf einem vieldimensionalen Datenraum übertragen können. Dadurch hätten sie auf sehr einfache und einheitliche Weise beides: 1. ein Integriertes Gebäudemodell mit ausreichender Kapazität und genügend inhaltlicher und struktureller Offenheit für alle Informationen eines Gebäudes über seine Lebenszeit und 2. ein entsprechend leistungsfähiges grafisches Interface.

In diesem Kapitel wird zuerst gezeigt, wie die verschiedenartigen Informationen, die bei der Kommunikation von Architekten, Ingenieuren und Nutzern während der Lebenszeit eines Gebäudes benutzt werden, auf eine einheitliche Art behandelt werden können. Der Hauptteil beschreibt die inhaltlichen, strukturellen und

ergonomischen Möglichkeiten einer 12-dimensionalen Arbeitsumgebung.

In diesem Kapitel liegt der Schwerpunkt auf den strukturellen Möglichkeiten des A4-Modell. Die benutzten Beispiele sind bewußt vielfältig. Sie werden ergänzt durch das Video im Anhang.

## Die Container

Für den Entwurf, die Konstruktion und den Betrieb von Gebäuden sind sehr viele verschiedene Medien erforderlich: Texte, Grafiken, Zeichnungen, Skizzen, Bilder, Videos, Faxe, Töne, Telefonate, Gespräche, Diskussionen, Konferenzen ... Es ist nicht das Ziel von A4, diese Medien selbst zu behandeln. Das ist die Aufgabe spezifischer Software. A4 kapselt diese verschiedenen Medien in einem Datencontainer ein. Ein Nutzer von A4 ist in der Lage die Daten eines Containers mit entsprechender Software *außerhalb* von A4 zu bearbeiten und dann *innerhalb* von A4 mit CAD-ähnlichen Techniken zu arrangieren.

## Der Datenraum

Die Datencontainer, die jede Art von Daten enthalten können, werden in A4 in einem 12-dimensionalen Datenraum angeordnet. Als eine erste Einführung in diese Idee stelle man sich die physikalischen Komponenten eines Gebäudes in einem 3-dimensionalen Datenraum angeordnet vor, wobei die Lage eines Bauteiles im Datenraum der wirklichen Lage im Gebäude entspricht. Jeder Container würde dann ein Bauteil des Gebäudes umschließen. Die Container könnten, einem LEGO-Baukasten vergleichbar, im 3D-Datenraum zu einem Gebäude zusammengesetzt werden (Bild 6). Da sich ein Nutzer des A4-Systems frei im Datenraum

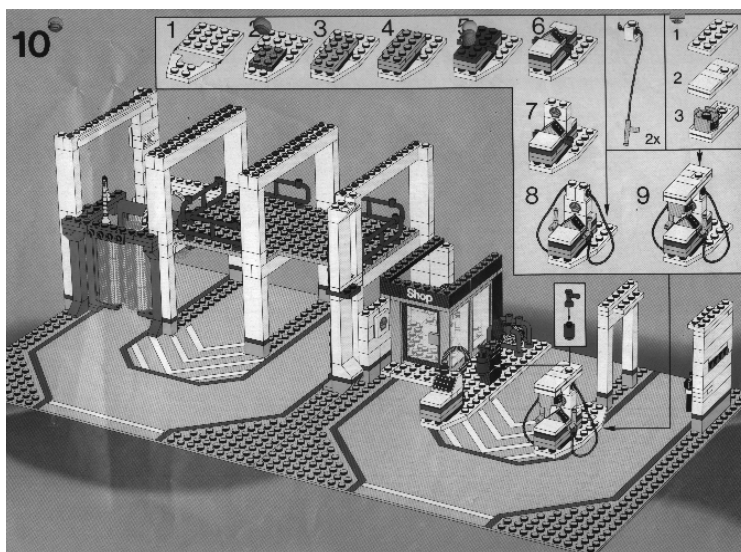


Bild 6 Das LEGO Baukastensystem

bewegen kann und da die Container wie ihre jeweiligen Bauteile dargestellt werden können, läuft ein Nutzer praktisch durch ein digitales Gebäude.

Neben einer Grafik kann ein Container auch Informationen in Form von Attribut-Wert-Paaren z.B. über Material, Gewicht, Preis oder Serviceintervalle enthalten. Dadurch bleibt das virtuelle Gebäude nicht nur ein grafisches Phänomen, sondern bekommt physikalische, ökonomische u.a. Qualitäten.

Besonders interessant ist die Möglichkeit, daß ein Container auch Software enthalten kann, sodaß er ein bestimmtes Verhalten haben kann. Auf diese Weise könnten - um diesem einfachen Beispiel zu folgen - Lichtschalter, eine Kontrolle der Raumtemperatur, Fassadensteuerungen oder Telefonapparate im Datenraum installiert werden.

Verläßt man die "Spielzeugwelt", dann müssen die Datencontainer auch nichtphysikalisch räumliche Komponenten eines Gebäudes enthalten können: Gruppen von Teilen, Abstraktionen, Absichten, Erläuterungen, Tutorien, Diagnosen, Konstruktionswerkzeuge, Simulationen ... Um auch diese Container im Datenraum anordnen zu können, werden in A4 neun zusätzliche Dimensionen für den Datenraum angeboten. Die Bewegung und das Arbeiten in einem 12-dimensionalen Gebäude sind ungewohnt. Die etablierten 3-dimensionalen Orientierungsmittel und Werkzeuge lassen sich aber auch auf die anderen Dimensionen übertragen.

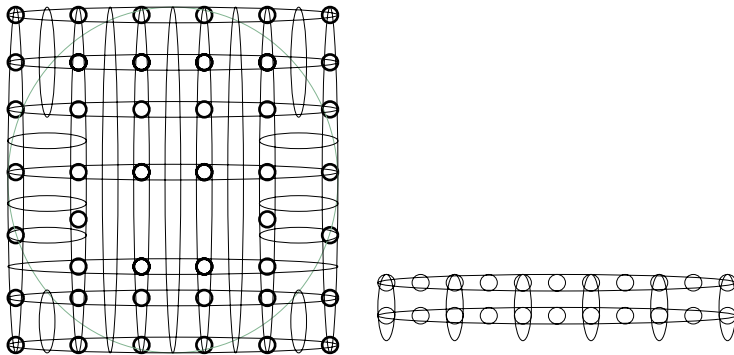
Im Folgenden werden die 12 Dimensionen des Datenraumes einzeln vorgestellt. Eine Zusammenfassung befindet sich in Tabelle 1 als Abschluß dieses Kapitels.

## Die Dimensionen x, y und z

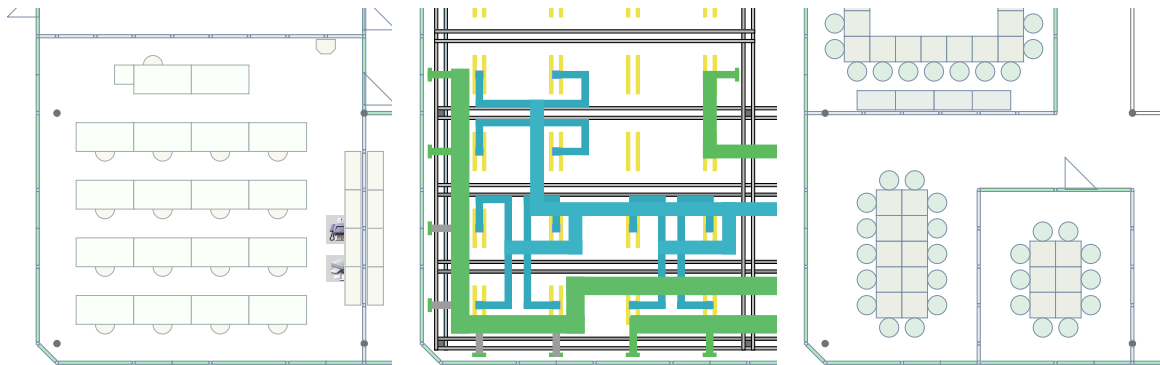
Auf diesen Achsen die Bereiche der Datencontainer in der Form "x, delta-x, y, delta-y, z, delta-z" markiert. Wie in der Einleitung beschrieben, ermöglichen die drei physikalisch-räumlichen Dimensionen, daß den Datencontainern direkt Baukomponenten entsprechen können, so daß - auf dem detailliertesten Darstellungsgrad- virtuelle Gebäude entstehen können (Bild 7a). Es sind aber auch abstraktere Darstellungen möglich: (Bild 8) zeigt eine Aufsicht und eine Ansicht der Konstruktion desselben Gebäudes. Besonders in der z-



**Bild 7** Links ist eine prototypische Konstellation von Datencontainern der Bausysteme MIDI (Konstruktion) und ARMILLA (Installationen) in einer x, y, z-Darstellung gezeigt. Die Datencontainer können unmittelbar einem gebauten Projekt entsprechen. In der Mitte und rechts ist ein entsprechendes Gebäude (Haller und Barth et.al., 1987) in der Außen- und Innenansicht dargestellt.



**Bild 8** Die Träger und Stützen eines zweigeschossigen Gebäudes, dargestellt als Skizze in  $x/y$  und in  $x/z$ .



**Bild 9** Dieses Bild zeigt eine einfache Bewegung auf der  $z$ -Achse. Links die Möblierung eines Seminarraumes in einer detaillierteren Darstellung als in Bild 8. Das mittlere Bild zeigt den Deckenhohlraum 3.00m höher. Rechts wird, nach einer weiteren Bewegung um 1.00m nach oben, die Möblierung im nächsten Geschoss gezeigt.

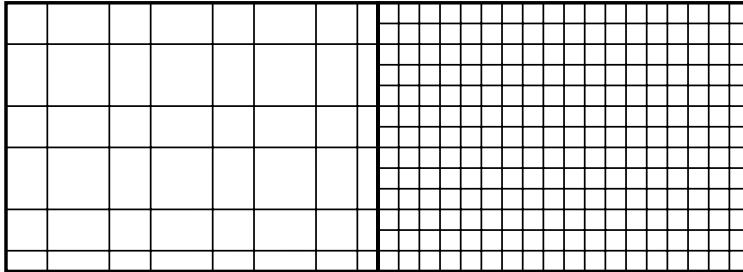
Achse sind Gebäude geschichtet, sodaß diese Achse wie Layer eines CAD-Systems benutzt werden kann (Bild 9).

Zur Bearbeitung von Containern in den Achsen des physikalischen Raumes ist nur ein 2- oder 3-dimensionaler Grafik-Editor erforderlich. Dazu genügen die folgenden Werkzeuge:

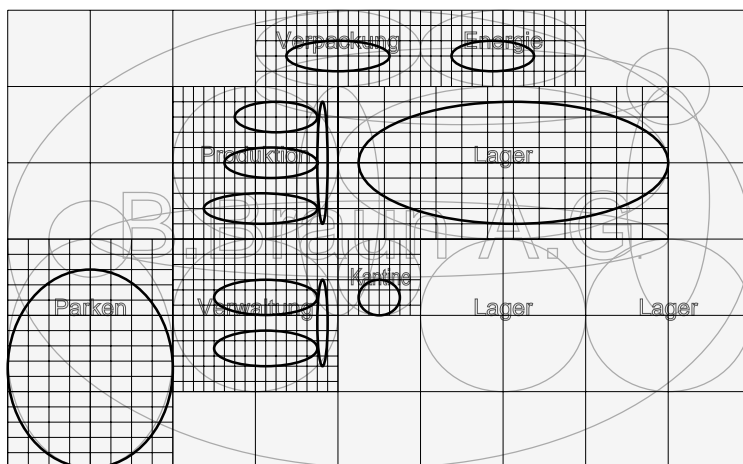
- für die Positionierung der Darstellungsrichtung und der Bearbeitungsflächen im 3-dimensionalen Raum (`vpoint`, `zoom`),
- für die Manipulation der räumlichen Koordinaten der A4-Container (`move`, `resize`, etc.),
- für das Editieren (`align`, `grid`, `template`).

Während ein A4 Editor für räumliche Zusammenhänge (`move`, `resize`, etc.) relativ primitiv gehalten werden kann, müssen die Funktionen `grid` und `template` der Erfahrung nach leistungsfähiger sein, als es bisherige CAD Systeme anbieten:

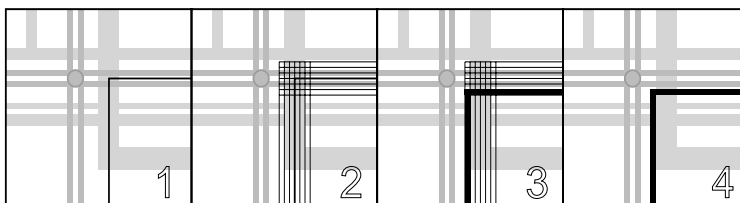
- `grid`: Besonders in den frühen Planungsphasen ist eine räumliche Strukturierung durch Raster sehr hilfreich, um die Container grob räumlich zu koordinieren. Die Container lassen sich jedoch nur selten in ein einheitliches und homogenes Raster einpassen, so daß in der Praxis der größte gemeinsame Teiler aller erforderlichen Raster gewählt wird (also unverhältnismäßig kleine Raster) (Bild 10). Durch eine Hierarchisierung können die Planungsraster auf verschiedenen Abstraktionsebenen schnell an örtliche Bedingungen angepaßt werden (Bild 10).



**Bild 10** Planungsraster sind oft nicht homogen (links). Würden die erforderlichen Rastermaße mit einem homogenen Raster erzeugt, so müßten sie unnötig detailliert sein und würden nur wenig Orientierungshilfe bieten (rechts).



**Bild 11** Planungsraster können schrittweise präzisiert werden. Dabei können sie sich an die lokalen Besonderheiten anpassen. Dieses Bild zeigt einen Übersichtsplan einer Fabrikanlage. Das übergeordnete Raster hat die Maße 64mx60m.



**Bild 12** Schablonen sind sehr wirkungsvolle Hilfsmittel, die es erlauben, in komplexen räumlichen Zusammenhängen schnell zu editieren.

- template: Besonders in den späteren und detaillierteren Planungsphasen sind Schablonen eine große Hilfe: In einer bestimmten Situation gibt es für die räumliche Anordnung der Container oft verschiedene Alternativen, die in einer 3-dimensionalen Schablone dargestellt werden können. Ein Nutzer kann durch einfaches Anwählen einer räumlichen Lage den Container 3-dimensional anordnen. Durch diese Technik ist es sehr oft möglich in einer nur 2-dimensionalen aber sehr übersichtlichen Darstellung alle 3 Dimensionen kontrolliert und schnell zu editieren (Bild 12).

Die Bedeutung des physikalischen Raumes geht weit über Layoutfragen von Grundrissen oder Stadtplänen hinaus. Es gibt z.B. Architekturschulen, die in der Baukonstruktion primär ein räumlich organisatorisches Problem sehen. Große Teile eines Architekturstudiums werden dafür benötigt, für bestimmte Anforderungen adäquate Räume zu formulieren. Auch umgekehrt befassen sich Baugeschichte oder Architekturtheorie oft mit der Frage, welche sozialen, ökonomischen, politischen oder technischen Bedingungen sich aus bestimmten räumlichen Situationen ableiten lassen.



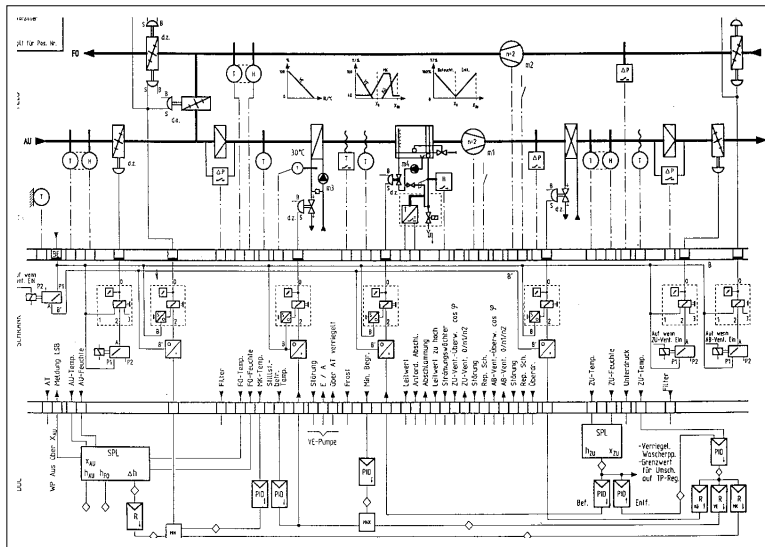


**Bild 13** Die am weitesten entwickelten Darstellungsmethoden sehr vielfältiger Zusammenhänge finden sich in der Kartographie (Dierke 69 p 2f)

Die Bedeutung räumlicher Strukturierungen für die Bauplanung mit einem geschichtlichen Überblick und extrem vielen Referenzen findet sich in Neuferts ständig aktualisierter Bauordnungslehre (Neufert 43).

Die am weitesten entwickelten Darstellungsmethoden räumlicher Zusammenhänge finden sich in der Kartographie. In der topographischen Karte in (Bild 13) z.B. sind Zonen, Markierungen, Beschriftungen, Farben usw. so gut aufeinander abgestimmt, daß sie eine erheblich höhere Informationsdichte erreichen als konventionelle CAD-Pläne oder die Darstellungen in A4 (vgl. (Bild 10)). Besonders interessant sind die Darstellungen von Zonen gleichen Luftdruckes, Höhen, Populationsdichten u.a. Diese flächigen Markierungen werden ergänzt durch lineare Darstellungen von Flüssen, Küstenlinien, Verkehrswegen u.a. Gleichzeitig werden auch Punkte von sehr kleinen Orten aber übergroßer Bedeutung dargestellt.

Es gibt Ausnahmen von der These, daß Planungsprobleme des Bauens sinnvoll räumlich bearbeitet werden können. (Bild 14) zeigt ein Beispiel für das Layout einer DDC (Direct Digital Control) einer Klimaanlage, die über mehrere Planungsstufen in einer konkreten räumlichen Situation der Komponenten der Klimaanlage verankert, logische Schaltbilder und letztlich Software Code bearbeitet.



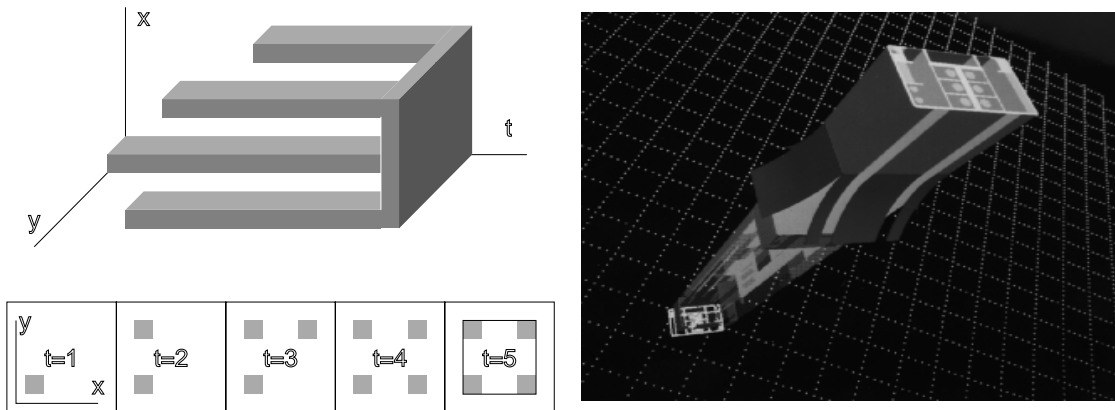
**Bild 14** Obwohl der Entwurf von Anlagen für die Gebäudesteuerung räumlich verankert ist, benutzt er dennoch funktionale und logische Beschreibungsformen auf mehreren Ebenen.

## Die Dimension Zeit

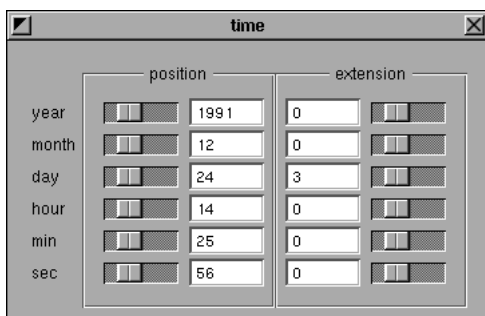
Auch auf der Zeitachse wird für jeden Datencontainer ein Bereich der Form "t, delta-t" markiert. Die Planungen und der Bauprozess entwickeln sich nicht nur räumlich, sondern auch über die Zeit. Bauablaufpläne, dynamische Gebäude, Lebenszyklus oder Facility Management deuten in diese Richtung. Eine starke zeitliche Komponente haben auch Gebäudesteuerungen, die zum Teil Echtzeitanforderungen stellen. Auch die ökonomischen, technischen oder sozialen Bedingungen ändern sich über die Zeit. Zusammen mit der geschichtlichen Kontinuität gebauter Umwelt spricht man von Morphologie besonders bei Städten (Muratori 59).

Moderne Software erschließt sich die Zeit mit Animationen, Simulationen, dem sogenannten "Cyber Space" oder Videospielen. Selbst konventionelle 3D-CAD-Systeme können benutzt werden, indem einfach eine der drei darstellbaren Achsen mit der Zeit belegt wird (Bild 15a). (Kohler 91a: WCSB) benutzt gezielt die Möglichkeiten eines computerunterstützten Films, um Planungen für das Renovieren von Altbauten transparenter zu machen.

Das einfachste und bekannteste Werkzeug für eine zeitliche Koordination ist ein Kalender, der in vielfacher Art als Software angeboten wird. Auch gibt es allgemein bekannte Software für das Project Management. Für A4 reicht aber oft ein einfacher Editor für die Zeitspanne eines A4 Containers (Bild 15). Bei komplexeren zeitlichen Koordinationsaufgaben kann der für die räumlichen Dimensionen beschriebene graphische Editor benutzt werden, indem eine der dargestellten räumlichen Dimensionen durch die Zeit ersetzt wird (durch die Funktion vpoint) (Bild 15b).



**Bild 15 a, b:** Konventionelle CAD Techniken können benutzt werden, um auch Zusammenhänge auf der Zeitachse zu editieren. Das Darstellungsprinzip (links) und ein komplexeres Beispiel (rechts) (V. Hovestadt 91).



**Bild 16** Ein einfacher Editor für die Position eines A4-Containers auf der Zeitachse.

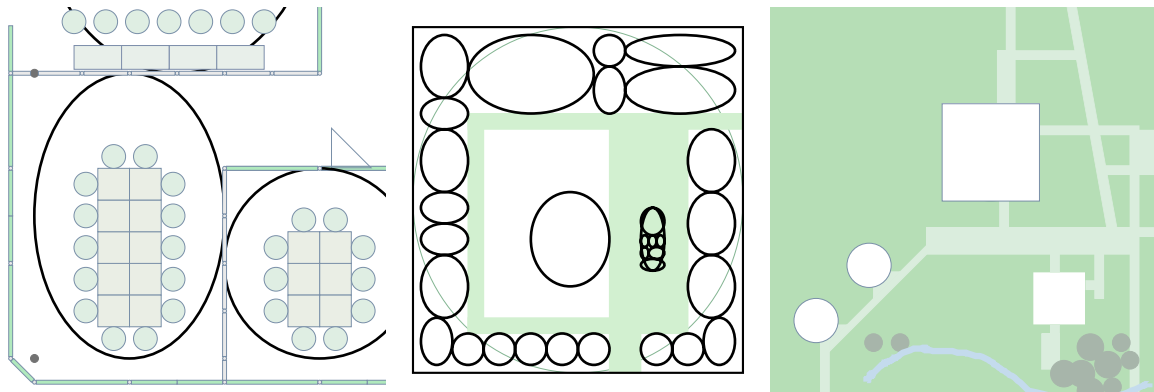
## Die Dimension Größenordnung

Wie es in den Begriffen Landesplanung, Regionalplanung, Städteplanung, Gebäudeplanung und Detailplanung zum Ausdruck kommt, lassen sich voneinander relativ unabhängige Planungen auf verschiedenen Größenordnungen beschreiben. Auch innerhalb eines Gebäudes gibt es Planungen verschiedener Größenordnung, die voneinander relativ unabhängig sind: die Gestaltung eines Türgriffes ist relativ unabhängig von der Gestaltung der Tür. Diese ist wiederum unabhängig von der Nutzung und Möblierung des Raumes, die wenig mit dem Grundrißlayout des Geschosses zu tun hat. Das Geschosß selbst ist wieder unabhängig von der Anordnung des Gebäudes auf dem Gelände. usw. (Bild 17)

Ein Zoom durch die Maßstäbe ist daher nicht kontinuierlich. Gute Beispiele sind im folgenden aufgeführt:

### *Powers of Ten*

Der Film "Powers of Ten" von Charles Eames (Eames 77), (Morrison 82), der einen großen Zoom durch die



**Bild 17** Drei Positionen auf der Größenordnungs-Achse: das Layout von Möbeln in einem Raum, von Räumen in einem Geschoss und von Gebäuden auf einem Grundstück. Obwohl die kleineren Planungen Teil der größeren Planungen sind, sind sie doch weitgehend unabhängig voneinander und jede für sich vergleichbar komplex.



**Bild 18** Der Film "Powers of Ten" von Charles Eames zeigt in einem großen Zoom alle dem Menschen bekannten Größenordnungen.

den Menschen bekannten Größenordnungen von  $10^{18}$  bis  $10^{-16}$  m zeigt (Bild 18).

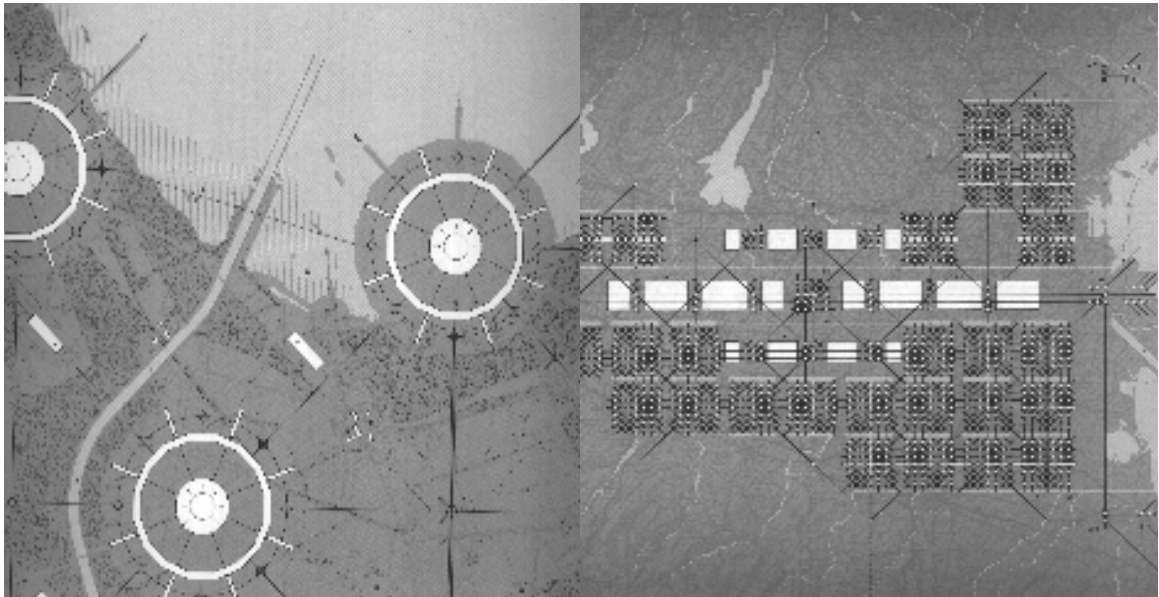
#### *totale stadt*

Die Studie "totale stadt" von Fritz Haller (Haller 75), die den menschlichen Lebensraum über verschiedene Ebenen bis hinauf zur globalen Ebene nur strukturieren kann, weil sie von der relativen gegenseitigen Unabhängigkeit der Beschreibungsebenen ausgeht (Bild 19).

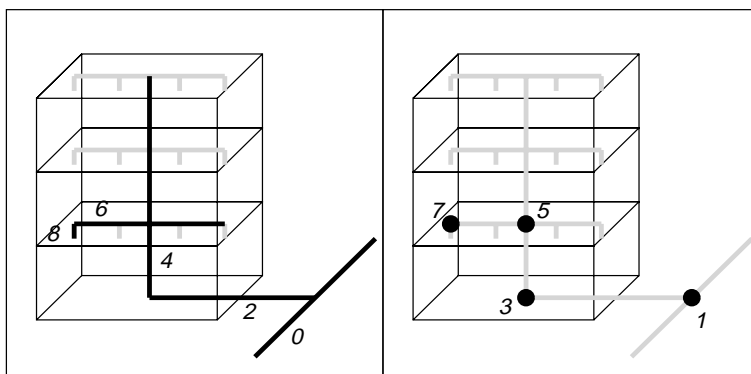
Wie besonders gut in der Morphologischen Schule Muratoris (Muratoris 59) diskutiert wird, sind Planungen in größeren Maßstäben öffentlicher, brauchen mehr Ressourcen und haben einen sehr langfristigen Effekt. Planungen in kleineren Maßstäben sind dagegen privater, spontaner und unverbindlicher.

Für einen Gebäudeentwurf kann das Phänomen der Größenordnung in zweierlei Hinsicht genutzt werden:

- Eine klare Gruppierung von Komponenten eines Entwurfes nach Größenordnungen ermöglicht gut voneinander abgegrenzte und überschaubare Teilplanungen.
- Die räumliche Gruppierung von Baukomponenten mit gleichen Standzeiten ist eine gute Gewähr für flexible Gebäude.



**Bild 19** Die Studie "totale stadt" von Fritz Haller zeigt in einem großen Zoom die Koordinationsprobleme des menschlichen Lebensraumes (Haller 75 p109 p83).



**Bild 20** Die Größenordnungen von Leitungssystemen in einem Gebäude

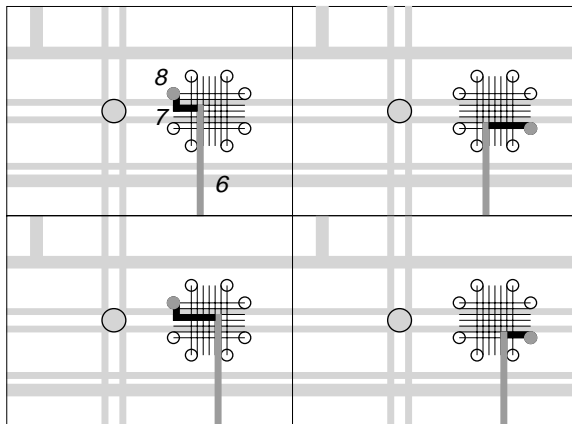
In den Beispielen dieser Arbeit werden folgende Größenordnungen benutzt (Bild 20):

*Größenordnung 2*

Alle A4-Container, die ein ganzes Gebäude beeinflussen, gehören in A4 zur Größenordnung 2. Ein Beispiel sind die Hausanschlußleitungen, mit denen ein Haus technisch ver- und entsorgt wird.

*Größenordnung 4*

Alle A4-Container, die ein ganzes Geschöß beeinflussen, gehören in A4 zur Größenordnung 4. Ein Beispiel sind die vertikalen Steigleitungen, mit denen ein Geschöß technisch ver- und entsorgt wird.



**Bild 21** Ein Beispiel aus ARMILLA zeigt, wie die relative Unabhängigkeit der Größenordnungen voneinander erreicht werden kann: Sowohl die Astleitungen (Größenordnung 6) als auch die Anschlüsse (Größenordnung 8) können in Grenzen (hier +/- 1.20m) verschoben werden, ohne daß diese Änderungen auf die Nachbarschaftsebene durchschlagen, weil die Zweigleitungen (Größenordnung 7) als Adapter jede Kombination gegenseitiger Lagen verbinden können.

#### Größenordnung 6

Alle A4-Container, die sich mit einem Ausschnitt eines Geschosses befassen (z.B. ein Raum), gehören in A4 zur Größenordnung 6. Ein Beispiel sind die horizontalen Leitungen in Deckenhohlräumen, die verschiedene Bereiche innerhalb der Geschosse technisch ver- und entsorgen.

#### Größenordnung 8

Als Beispiel für A4-Container der Größenordnung 8 können die vertikalen Leitungen in einer Installationswand gelten, die die verschiedenen Geräte technisch ver- und entsorgen.

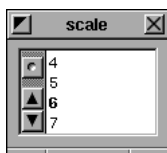
#### Größenordnung 10

Die Geräte selber haben in A4 die Größenordnung 10.

An diesem Beispiel zeigt sich, daß Planungen und Änderungen in den verschiedenen Größenordnungen verschieden aufwendig und langfristig sind. Während ein Gerät (Größenordnung 10) relativ spontan und ohne große Auswirkung auf das Gebäude verschoben oder ausgetauscht werden kann, ist der Umbau oder das Verlegen einer Hausanschußleitung (Größenordnung 2) sehr viel kostspieliger und hat tiefgreifende Folgen für ein Gebäude.

Damit die Planungen auf den beschriebenen Größenordnungen in den Beispielen dieser Arbeit die geforderte Unabhängigkeit voneinander bekommen, gibt es Adapter zwischen den Größenordnungen, die mit 3, 5, 7 und 9 bezeichnet werden (Bild 20), (Bild 21). Dabei regelt z.B. Größenordnung 7 den Übergang von einer horizontalen Verteilung im Deckenhohlraum in den vertikalen Leitungen in einer Installationswand.

Zum Editieren der Größenordnung bietet sich ein einfaches Werkzeug an (Bild 22).



**Bild 22** Ein einfacher Editor für die Position eines A4-Containers auf der Größenordnungsachse.

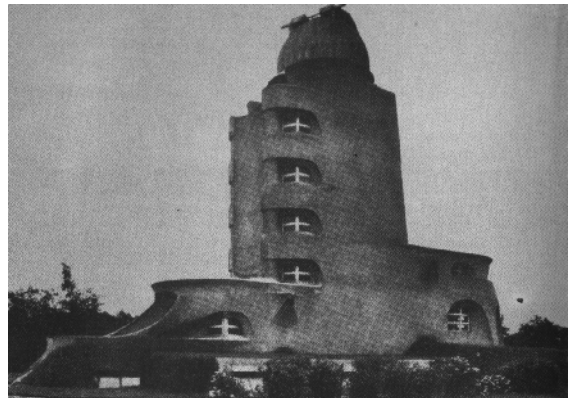
# Die Dimension Auflösung

Eine Planung beginnt in der Regel mit groben und vagen Beschreibungen und wird mit der Zeit detaillierter (Bild 23). In der Anfangsphase einer Planung sind Skizzen, unverbindliche Gespräche, Brainstorming etc. wichtig. Sie endet dagegen mit der möglichst genauen Festlegung von Konstruktionen, Funktionen und Leistungen. Die hier beschriebene Auflösung einer Planung ist klar von der oben beschriebenen Größenordnung zu unterscheiden, denn die Auflösung einer Planung entwickelt sich sowohl in der Regionalplanung, der Städteplanung, der Gebäudeplanung als auch in der Detailplanung vom Groben zum Feinen.

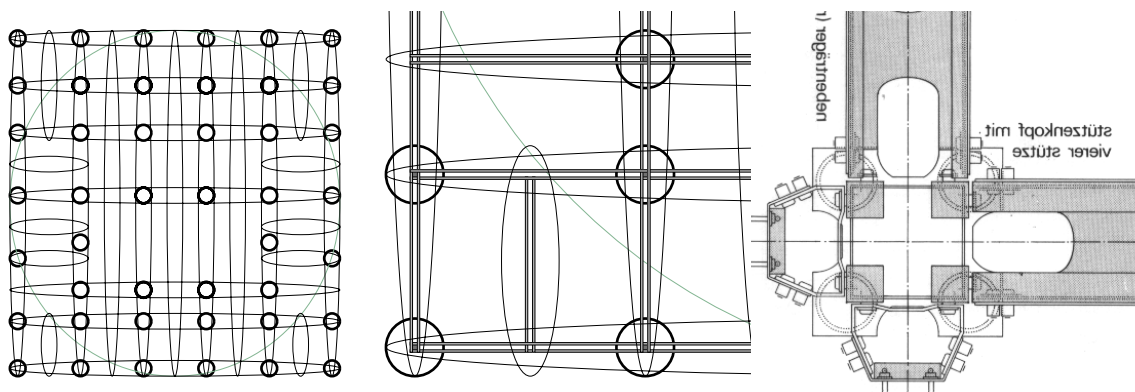
Um dieser Dimension des Planungsprozesses gerecht werden zu können, schlägt A4 eine Unterscheidung in drei grundsätzlich verschiedene Auflösungen vor (Bild 24):

## Zone

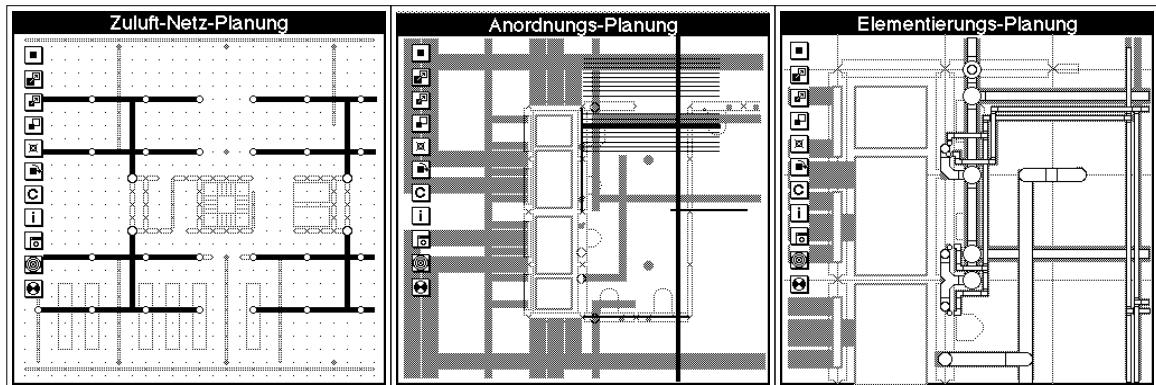
Zonen bezeichnen räumliche Bereiche, die übergroß sind und die sich deshalb gegenseitig überlagern und durchdringen können. Sie werden von A4 in der Regel mit Ellipsen oder Freiformflächen dargestellt.



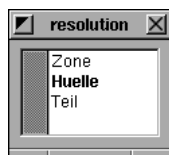
**Bild 23** Eine Skizze Mendelsohns zum Einsteinturm und das realisierte Gebäude (1920) (Benevolo 78, Bd.2 pp 76f)



**Bild 24** Drei verschiedene Auflösungsstufen der Konstruktion eines Gebäudes, von der Skizze (Zonen) über die detaillierte Übersicht (Hüllen) zum Detail (Teil).



**Bild 25** Ein Beispiel aus ARMILLA kann die Bedeutung der Auflösungsdimension für den Gebäudeentwurf zeigen: in der Linienleitungsplanung (Bild a: entspricht den Zonen) wird in einem ersten Schritt das grobe Layout eines Leitungsnetzes festgelegt. In der folgenden Astleitungsplanung wird innerhalb der Zone einer Linienleitung die Hülle für eine Astleitung gesucht (Bild b). In einem letzten Planungsschritt werden innerhalb der Astleitungen die physikalischen Teile angeordnet (Bild c). Durch diese Gliederung der Planung in Planungsschritte verschiedener Genauigkeit ist keiner der Planungsschritte übermäßig komplex.



**Bild 26** Ein einfacher Editor für die Position eines A4-Containers auf der Auflösungsachse

### Hülle

Hüllen bezeichnen räumliche Bereiche, die minimal groß sind, exklusiv Raum für eine bestimmten Zweck reservieren und deshalb untereinander kollidieren. Sie können mit Rechtecken oder Polygonen dargestellt werden.

### Teil

Teile bezeichnen Container, die nicht weiter bearbeitet werden sollen. Sie können deshalb so dargestellt werden, wie sie sind, z.B. mit einem Photo. Teile sind nach dieser Auffassung nicht nur Details. Teile sind auch z.B. Bebauungsgrenzen, schon existierende Gebäude oder das Gelände. Teile sind also ganz allgemein Zusammenhänge, die man nicht weiter bearbeiten will, kann oder darf und die deshalb als Konstanten gelten.

Eine Planung wird in der Regel mit sehr stark überdimensionierten Zonen beginnen, die im Verlaufe der Planung schrittweise bis auf Hüllen präzisiert werden. Eine Planung schließt mit den Teilen innerhalb der Hüllen ab (Bild 25).

Ein sehr interessantes Problem der Auflösungs-Achse ist die Koordination der Maße: Die Teile, z.B. ein Ziegel, eine Treppenstufe, ein Stahlprofil, aber auch die Umgebung eines Gebäudes, wie die Topographie oder Anschlußpunkte der Erschließung, haben in der Regel festgelegte und nicht veränderbare Maße. Sie haben in A4 daher auf der Auflösungs-Achse die Position Teile. Die Zonen sind dagegen übergroß und haben nur relative Bezüge. Wird nun über die Zonen schrittweise eine System von relativen räumlichen Abhängigkeiten aufgebaut, und wird die Planung soweit präzisiert, daß die Zonen mit ihrem relativen Maßsystem in das absolute Maßsystem der Hüllen und Teile überführt werden sollen, so kommt es oft zu Konflikten zwischen den beiden Maßsystemen.

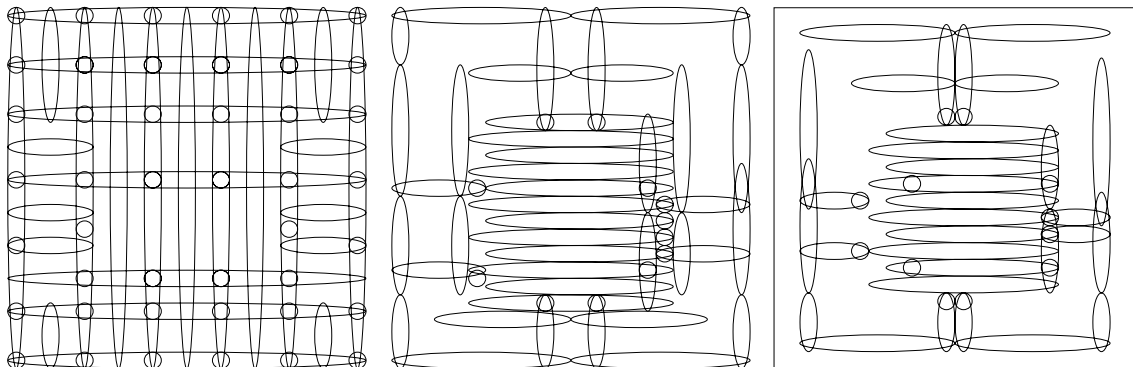


# Die Dimension Teilsystem

Entsprechend den verschiedenen traditionellen Gewerken oder den verschiedenen technischen Systemen eines Gebäudes kann der Bauprozess in verschiedene Teilsysteme gegliedert werden. Sie durchziehen weite Bereiche des Bauprozesses und werden von einzelnen Fachleuten betreut. Beispiele sind Rohbau, Möblierung, Beleuchtung, elektrische Versorgung, Klimatechnik, Akustik, Statik, etc.

Weil sich diese Gliederung so stark an die traditionellen Arbeitsteilung im Bauprozess anlehnt, wird sie oft als ausschließliches Modell für Concurrent Design benutzt (CIB 91, Pohl 88: ICADS et al.), um komplexe Planungsaufgaben in kleinere Einheiten zu gliedern. Es wird dabei übersehen, daß die einzelnen Teilsysteme selbst noch eine unüberschaubare Komplexität besitzen. Eine weitere Gliederung ist daher erforderlich. Dann tritt aber häufig der Fall ein, daß sich verschiedene Teilaufgaben traditioneller Fachplanungen so überlagern, daß sie zusammengefaßt werden können. Dabei verwischen die Zuständigkeiten der einzelnen Fachplaner, so daß bei einem integrierten System die traditionellen Trennungen in Gewerke nicht mehr so tragfähig sind. Deshalb werden sie in A4 nur als eine von vielen Dimensionen des Baumodells benutzt.

In A4 wird vorgeschlagen, Teilsysteme der Erschließung, Konstruktion, Möblierung, Zuluft, Abluft, Akustik, Fassade ... zu unterscheiden. Sie sind, wie die Morphologie, der Maßstab oder die Auflösung, sehr stark von der konkreten räumlichen Situation und deren besonderen Planungsproblemen abhängig. Deshalb wird auch hier als Editor eine Liste aller an einem bestimmten Ort etablierten Teilplanungen vorgeschlagen (Bild 28).



**Bild 27** Drei Positionen auf der Teilsystem-Achse: links eine Skizzendarstellung der Konstruktion eines Bürogebäudes, in der Mitte eine Skizze der Abluftleitungen im Deckenhohlraum, rechts die entsprechende Skizze des Zuluftsystems.



**Bild 28** Ein einfacher Editor für die Position eines Containers auf der Teilsystemachse

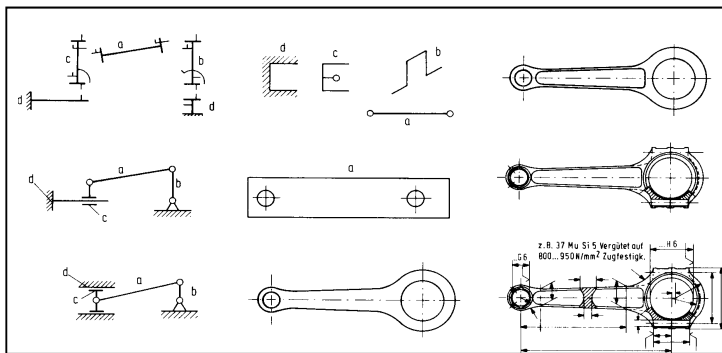
# Die Dimension Morphologie

Eine weitere Dimension kann als Morphologie bezeichnet werden. Hiermit ist das Phänomen gemeint, daß eine Planung auf verschiedene Arten nacheinander oder sogar gleichzeitig beschrieben wird. Roth (Roth 82) und Pahl (Pahl 77) beginnen z.B. in ihrer Konstruktionslehre für den Maschinenbau mit einer funktionalen Beschreibung, entwickeln daraus eine strukturelle Beschreibung, dann eine Kontur und schließlich die vollständigen Fertigungsunterlagen (Bild 29). David (David 87), Rivero (Rivero 77), Rittel (Rittel 70), Bedell (Bedell 93) und viele andere beginnen z.B. den architektonischen Entwurf mit funktionalen Anforderungen, die sie in Teilanforderungen gliedern, bis räumliche oder technische Lösungen assoziiert werden können.

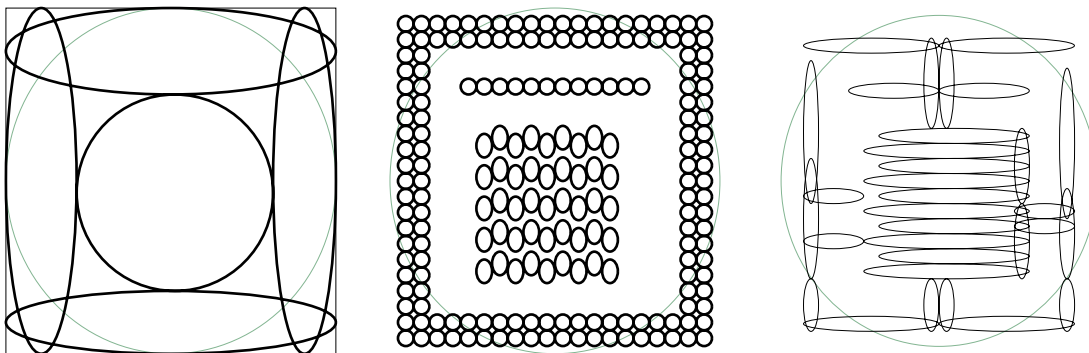
Für A4 werden anfänglich drei einander ergänzende Beschreibungen vorgeschlagen, die in einem anderen Zusammenhang anders interpretiert, erweitert oder ersetzt werden können (Bild 30):

## Nutzung

Nutzungen (NU) bezeichnen Anforderungen an einen räumlichen Bereich,



**Bild 29** verschiedene Beschreibungsformen eines Maschinenteils (Roth 82 p 28)



**Bild 30** Ein Beispiel für drei benachbarte Positionen auf der Morphologie-Achse: links die Anforderungen an das Zuluftsystem eines Bürogebäudes, das entlang den Fassaden anders ausgelegt sein muß als im Innern des Gebäudes (Nutzung). In der Mitte sind die dafür notwendigen Zuluft-Auslässe dargestellt (Ausstattung). Rechts sieht man die für die Auslässe notwendigen Zuluft-Leitungen (Erschließung). Alle Darstellungen sind von der Auflösung einer Skizze.

### Ausstattung

Ausstattungen (AU) bezeichnen räumliche Bereiche baulicher und/oder technischer Ausstattungen um die NU zu ermöglichen.

### Erschließung

Erschließungen (ES) bezeichnen räumliche Bereiche baulicher und/oder technischer Systeme oder Subsysteme die die AU ver- oder entsorgen.

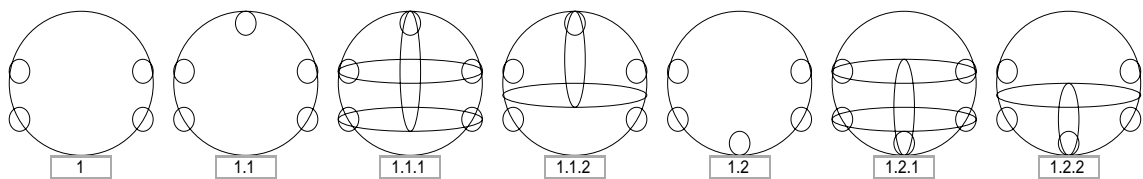
In konkreten Projekten sind diese Unterscheidungen oft schwer voneinander abzugrenzen und können daher oft nicht sinnvoll und durchgängig benutzt werden. Deshalb ist mit einer Vielzahl oft konkurrierender Beschreibungen zu rechnen, die sich aber, da sie u.a. räumlich voneinander getrennt werden können, nicht gegenseitig ausschließen müssen. Als ein Werkzeug für die morphologische Dimension bietet sich eine einfache Liste der verschiedenen, an einem bestimmten Ort bereits etablierten Beschreibungen an. Diese Liste kann erweitert und bearbeitet werden (Bild 31).



**Bild 31** Ein einfacher Editor für die Position eines Containers auf der Morphologie-Achse.

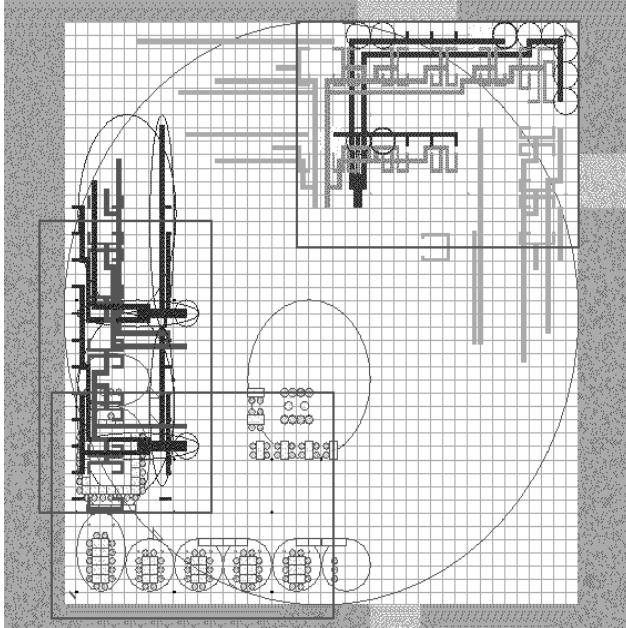
## Die Dimension Alternativen

Wenn ein Gebäude neu entworfen oder umgeplant wird, arbeiten die Planer an verschiedenen alternativen Fragestellungen und Lösungen um einen besonders guten Entwurf zu entwickeln. Die dabei entstehenden Hierarchien von Alternativen können einfach auf einer weiteren Achse des Datenraumes abgebildet werden. Das Beispiel in Bild 32 zeigt eine X,Y Sicht auf ein Zuluft-Netz mit der Auflösung Zone aus sieben verschiedenen Positionen der Alternativen-Achse.



**Bild 32** sieben verschiedene Positionen auf der Alternativen-Achse.

## Die Dimension Nutzer



**Bild 33** *Drei Verschiedene Nutzer, repräsentiert durch die drei großen Rechtecke, arbeiten gleichzeitig an verschiedenen Stellen in einem Gebäude.*

Diese Achse des Datenraumes ist die Basis für Multiuser Umgebungen im Datenraum. Das Beispiel in Bild 33 zeigt die Arbeitsumgebungen von drei Nutzern in einem Gebäude. Sie können in A4 genau dann kommunizieren und zusammenarbeiten, wenn sich ihre Arbeitsumgebungen überlagern, wie man es bei den beiden Nutzern unten links sehen kann. Es ist offensichtlich, daß Probleme wie Transaktionsmanagement, Caching oder der Verkehr auf dem Netzwerk durch die räumliche Organisation der Container und der Nutzer entschärft werden.

## Die Dimension Timetag

Undo, Backtracking, Truth Maintenance usw. arbeiten auf der Timetag-Achse, auf der die Zeitpunkte des Erzeugens, des Modifizierens und des Löschens von Containern aufgezeichnet werden. Weil diese Achse sehr schnell große Datenmengen erzeugt, muß ein "Garbage Collector" installiert werden.

# Die Dimension Komposition

Manche Informationen können nicht sinnvoll ausschließlich über die räumlichen Position der Container koordiniert werden. Möchte man z.B. eine Gruppe von Containern zu einer Problemstellung zusammenfassen, so kann der Problem-Container zwar die anderen Container in x, y, z und t umfassen, kann jedoch nicht von einer Kollision auch auf den anderen Achsen ausgehen. Die Kompositionsachse bietet einen Ausweg: Container kollidieren auf jeden Fall dann, wenn sie auf der Kompositions-Achse die gleiche Position haben.

# Die Dimension Meta

Wenn Informationen über andere Informationen abgelegt werden sollen, dann können sie auf dieser Achse eine besondere Position bekommen. Die Default-Position ist Domain. Auf ihr werden die Informationen angeordnet, die unmittelbar das Gebäude betreffen. Informationen, die auf anderen Positionen dieser Achse angeordnet sind, betreffen nicht das Gebäude, sondern Informationen auf der Position Domain dieser Achse, also nur mittelbar das Gebäude. Dabei kann ein Container nur dann Informationen über einen anderen Container beinhalten, wenn er entweder mit ihm kollidiert, oder auf der Kompositions-Achse die gleiche Position hat (Die Begriffe Nachbarschaft und Kollision werden in den folgenden Abschnitten diskutiert) Hier sollen einige Positionen auf der Meta-Achse vorgestellt werden. Die Aufzählung ist, wie bei den anderen Achsen auch, weder bindend noch vollständig.

## *domain*

Die Default Position. Sie bezeichnet Container, die unmittelbar das Gebäude betreffen.

## *attribute*

Jeder Container kann strukturierte Informationen in Form von Attribut-Wert-Paaren enthalten (siehe Abschnitt Wissensrepräsentation). Um Redundanzen zu vermeiden besteht die Möglichkeit, in übergeordneten Containern Attribute zusammenzufassen.

## *rule*

Ein Container kann auch Wissen in Form von Regeln beinhalten (siehe Abschnitt Wissensrepräsentation). Hier besteht, wie bei den Attributen, die Möglichkeit, die Regeln in übergeordneten Containern zusammenzufassen.

## *constraint*

Constraints sind eine andere Form der Wissensrepräsentation, die in Containern abgelegt werden und in übergeordneten Containern zusammengefaßt werden können (siehe Abschnitt Wissensrepräsentation).

## *inference*

Die verschiedenen Wissensrepräsentationen (frames, rules, constraints) können mit verschiedenen Inferenzmechanismen bearbeitet werden. So wie die Attribut-Container, die nur dann Informationen über ihren Domain-Containern repräsentieren können, wenn sie mit ihm kollidieren oder eine gemeinsame Position auf

der Kompositions-Achse haben, so können auch die Inferenz-Container nur unter diesen Bedingungen über Domain-, Attribute-, Rule- oder Constraint-Container schließen.

*task*

Hier werden die Aufgabenstellungen einer Planung formuliert. Sie können z.B. in Zusammenhang mit Case-Based Reasoning stehen, wie z.B. in (Kolodner 92) diskutiert, oder Inferenzen steuern, wie z.B. in (Walter et al.92) oder (Wielinga et al. 92).

*problem*

Hiermit können Container zu Problemen zusammengefaßt werden, z.B. als Bestandteil eines Falles in Case-Based Reasoning.

*solution*

Hiermit können Container als Lösung z.B. für Probleme zusammengefaßt werden.

*operator*

Hiermit können Bearbeitungsschritte die z.B. nötig waren ein Problem zu lösen, zusammengefaßt werden.

*case*

Cases fassen üblicherweise Aufgaben, Probleme, Lösungen, Erläuterungen u.a. zusammen (Kolodner 92).

*catalog-object*

Container müssen nicht notwendigerweise Gebäude betreffen. Sie können auch z.B. Teile eines Kataloges oder Lehrbuches sein.

## Die Skalen

*In diesem abschließenden Teil werden die Dimensionen des Datenraumes zusammengefaßt.*

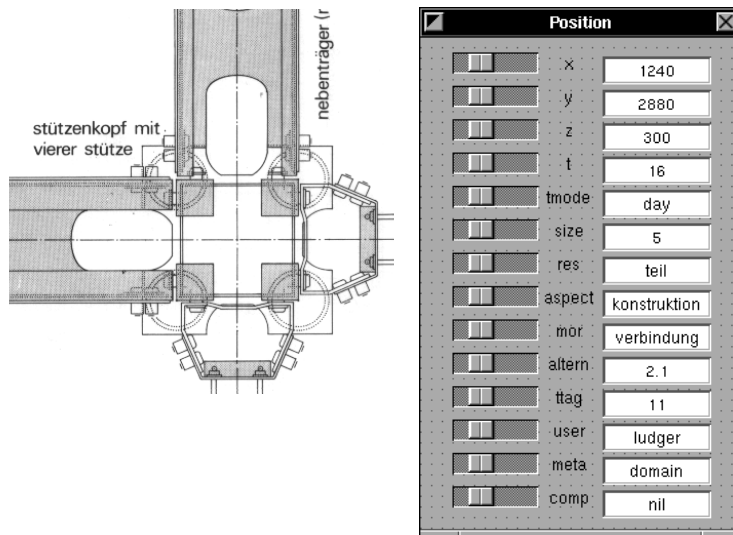
Die Container erhalten im vieldimensionalen Datenraum eine **Position**. Ein Container hat in diesem Raum in jeder Dimension bestimmte **Nachbarn**. Alle Nachbarn eines Containers werden als seine **Umgebung** bezeichnet. Diese Begriffe sind abhängig von den Skalierungen der einzelnen Achsen des Datenraumes. (Tabelle 1) gibt einen Überblick.

Achse	Anordnung	Position	Kollision	Nachbarschaft
<b>x,y,z,t</b>	kontinuierliche Skala	ein Bereich der Skala	Überlagerung der Positionen	Überlagerung der Positionen
<b>Größenordnung</b>	geordnete Liste	ein Element der Liste	gleiches Element	beide Nachbarelemente der Liste

**Table 1**

Achse	Anordnung	Position	Kollision	Nachbarschaft
<b>Auflösung</b>	geordnete Liste	ein Element der Liste	gleiches Element	beide Nachbarelemente der Liste
<b>Teilsystem</b>	ungeordnete Menge	ein Element der Menge	gleiches Element	jedes Element der Menge
<b>Morphologie</b>	ungeordnete Menge	ein Element der Menge	gleiches Element	jedes Element der Menge
<b>Alternativen</b>	Baum	ein Knoten	Element des Teilbaums	benachbarte Knoten ersten und zweiten Grades
<b>Timetag</b>	kontinuierliche Skala	ein Bereich der Skala	Überlagerung der Positionen	Überlagerung der Positionen
<b>Nutzer</b>	ungeordnete Menge	ein Element der Menge	gleiches Element	jedes Element der Menge
<b>Meta</b>	ungeordnete Menge	ein Element der Menge	gleiches Element	jedes Element der Menge
<b>Komposition</b>	ungeordnete Menge	mehrere Elemente der Menge	mindestens ein gleiches Element	jedes Element der Menge

Table 1



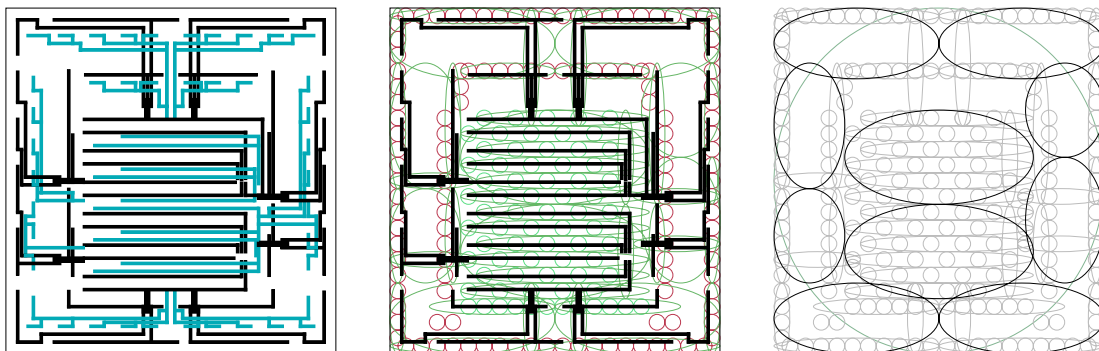
**Bild 34** Ein Stützenkopf auf dem detailliertesten Beschreibungslevel und ein typisches Positionspanel von A4. Die Position des Stützenkopfes im 12-dimensionalen Datenraum ist:  $x: 1240, dx: 40, y: 2880, dy: 40, z: 300, dz: 120, t: 16.9.1993.14.30, dt: inf$ , Größenordnung: 5, Auflösung: teil, Teilsystem: konstruktion, Morphologie: verbindung, Alternativen: 2.1, Timetag: 11.2.1993.9.13.56.45, Nutzer: ludger, Meta: domain, Komposition: nil.

# 4 Die Navigation im A4-Modell

Eine Planung besteht aus dem Installieren, Bewegen und Löschen von Containern im Datenraum. Sie umfaßt, entsprechend den Dimensionen des Designraumes, u.a. räumlich-zeitliche und funktionelle Positionen und Bewegungen, es können Skizzen in präzise Darstellungen überführt werden, es können Typen aus Beispielen entwickelt werden, oder es können, entsprechend den Erfahrungen manueller Tätigkeiten, Automatismen programmiert oder Planungsstrategien entwickelt werden.

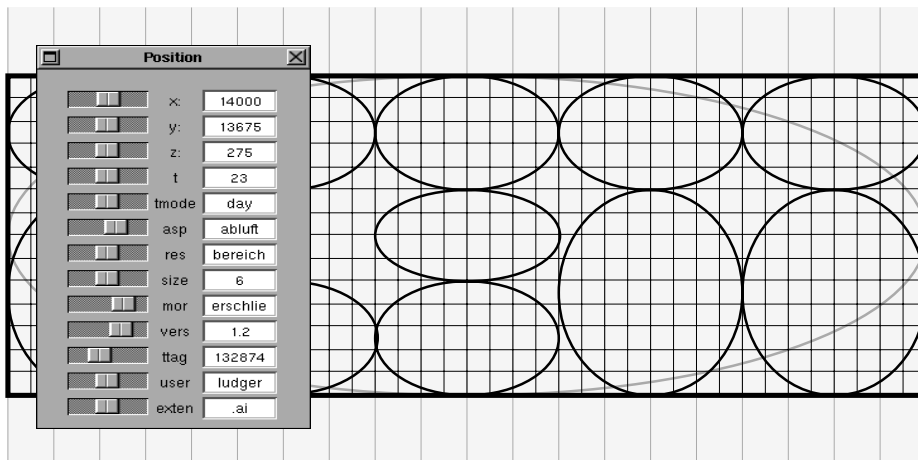
Dabei muß zwischen einer makroskopischen und einer mikroskopischen Sicht des Designprozesses unterschieden werden. Während auf der makroskopischen Ebene klare Entwicklungslinien und sogenannte Meilensteine beschrieben werden können, sind auf der mikroskopischen Ebene Hypothesen, Tests und Iterationen dominant (Müller 89).

Die Bewegungen dienen in A4 dazu, mit der besonderen Komplexität der Daten im Datenraum umzugehen: das A4-Modell schlägt vor, daß sich die Planer von Container zu Container bewegen können und jeweils nur die überschaubar komplexe, lokale Umgebung eines Containers beobachten und bearbeiten müssen. Um auch andere Bereiche des Designraumes zu beobachten und zu bearbeiten, muß der Planer die Position eines anderen Containers einnehmen. Er muß also, damit seine Planungsumgebung nicht unüberschaubar komplex wird, bestimmte Informationen verlassen, um neue Informationen in seine Planungsumgebung aufnehmen zu können. Dabei kann zwischen relativen, absoluten und assoziativen Schritten einer Bewegung unterschieden werden: relative Schritte verlaufen entlang der Nachbarschaften der Container, absolute Schritte benutzen gespeicherte Raumkoordinaten oder eindeutige Namen der Container, assoziative Schritte suchen mit Verfahren des pattern recognition oder case-based reasonings eine der aktuellen Arbeitsumgebung vergleichbare andere Arbeitsumgebung an anderem Ort im Datenraum. Das letzte Verfahren ist noch sehr experimentell. Bild 35 gibt ein Beispiel für eine relative Bewegung in einer Arbeitsumgebung mit einer üblichen Menge dargestellter Container.

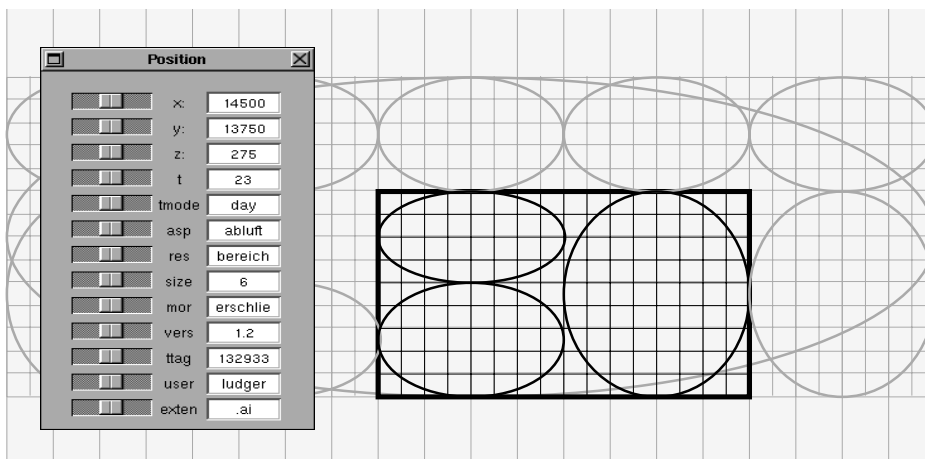


**Bild 35** Drei verschiedene Positionen eines Nutzers im Datenraum mit einer durchschnittlichen Menge dargestellter Container. Links die Hüllen des Klimasystems eines Bürogebäudes in einem Deckenhohlraum. In der Mitte, nach einer Bewegung auf der Auflösungs-Achse, zusätzlich die Zonen des Abluft-Systems im Deckenhohlraum. Rechts sind die Hüllen ausgeblendet.





**Bild 36** Die Position und die Umgebung eines Nutzers im Datenraum. Angezeigt sind die räumliche Position (das dick gezeichnete Rechteck im großen Fenster), die den gesamten Deckenhohlraum umfaßt, und die Position "Erschließung" auf der morphologischen Achse, die in diesem Fall die Netze für Abluft (abl-esb6) anzeigt.

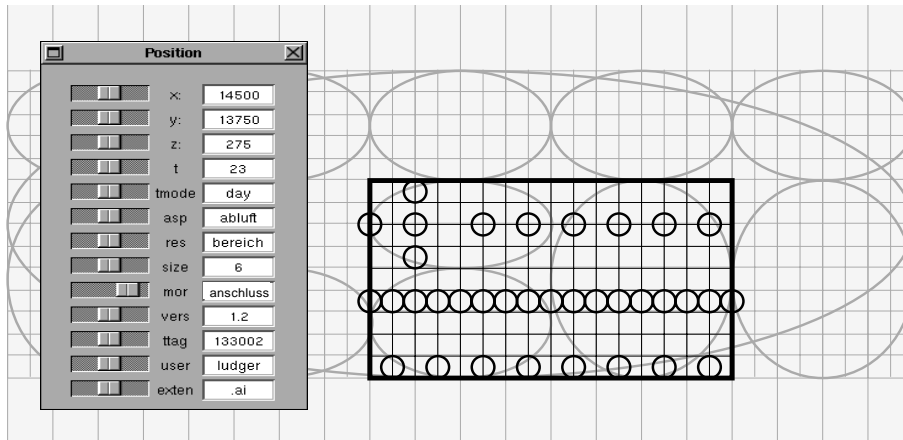


**Bild 37** Der Nutzer konzentriert sich durch eine Bewegung auf den Achsen x und y auf einige besondere Abluftnetze, weil diese der Regelmäßigkeit der Gesamtanordnung widersprechen.

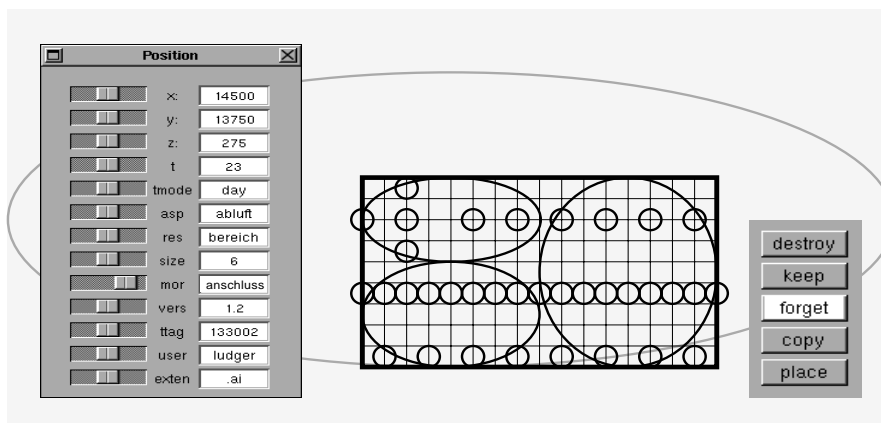
Im folgenden sollen die elementaren Prinzipien und Operatoren einer Bewegung im Datenraum beschrieben werden, indem sie mit Kommandos des UNIX-Betriebssystem verglichen werden:

Ein Planer befindet sich, wie jeder Container, an einem bestimmten Ort im Datenraum. Dieser Ort wird mit den in den Dimensionen des Datenraums beschriebenen Werkzeugen besonders hervorgehoben. Als Beispiele sollen hier der Editor für die räumliche Dimension und die Morphologie dienen (Bild 36). Diese Anzeige der Position eines Nutzers ist mit dem UNIX-Kommando `pwd` vergleichbar, das die augenblickliche Position im UNIX-Filesystem anzeigt.

Gleichzeitig bieten die Werkzeuge des Datenraumes dem Nutzer aber auch einen Überblick über seine unmittelbare Umgebung. Bild 36 zeigt neben der momentan gewählten Beschreibungsform eines Problems



**Bild 38** Eine Bewegung in die Position "anschluss" auf der Morphologie-Achse soll Aufschluß über die Ursache des unregelmäßigen Ablufnetzes geben.



**Bild 39** Der Nutzer löscht die Darstellungen nicht mehr benötigter Container seines Weges im Datenraum mit dem Kommando `forget`. Dadurch bleibt seine Oberfläche übersichtlich.

des Nutzers (schwarz) auch andere bis dahin etablierte Container an (grau). Auf dem Plan sind Container in räumlicher Nachbarschaft dargestellt. Auf dem Fenster der morphologischen Achse sind alle an der Position des Nutzers bereits etablierten Beschreibungsformen angezeigt. Als Vergleich für diese Werkzeuge, die die Umgebung eines Nutzers an einer Position anzeigen, bietet sich das UNIX-Kommando `ls` an.

Entsprechend den vielen Dimensionen des Datenraumes bieten sich für einen Nutzer sehr vielfältige Möglichkeiten, sich auf benachbarte Orte im Datenraum zu bewegen. Vergleichbar dem UNIX-Kommando `cd` kann sich ein Nutzer auf den verschiedenen Achsen des Datenraumes bewegen: Mit dem Kommando `(move-to '(x 100 dx 100 y 200 dy 120 z 0 dz 30))` konzentriert sich ein Nutzer auf einen Ausschnitt mit besonderer Unregelmäßigkeit (Bild 36). Eine weitere Bewegung auf der morphologischen Achse mit dem Kommando `(move-to 'anschluss)` (*'anschluss ist eine Spezialisierung einer 'ausstattung*) führt näher an die Ursachen der Unregelmäßigkeiten heran (Bild 38). Der Befehl `move-to`

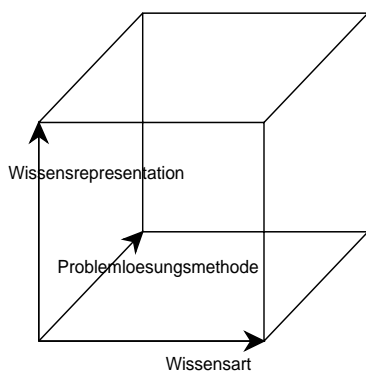
läßt sich sinngemäß auch auf die anderen Dimensionen des Datenraumes übertragen.

Kennzeichnend für diese Bewegungen ist es, daß sie in der Regel nicht auf absoluten Koordinaten oder auf neuen Bezeichnungen auf den Achsen des Datenraumes basieren, sondern sich an bereits etablierten Containern orientieren. Für eine Bewegung im physikalischen Raum bedeutet dies, daß das `move-to` Kommando etwa so formuliert wird: bewege dich auf den x, y und z Koordinaten dieses Containers.

Bei diesen Bewegungen ist es wichtig, daß die Navigationswerkzeuge ein "Kurzzeitgedächtnis" haben, d.h. daß sie die Darstellung der alten Umgebungen beibehalten, bis sie durch ein explizites `forget` Kommando gelöscht werden (Bild 39).

Das Wechselspiel der Kommandos `move-to` und `forget` kann in Verbindung mit einer guten graphischen Repräsentation der Container sehr intuitiv benutzt werden. Es wäre sogar - bei einer optimierten Implementation - einem Videospiel vergleichbar, was für die Nutzerfreundlichkeit und die Akzeptanz des Systems besonders wichtig ist.

# 5 Methoden der KI im A4-Modell

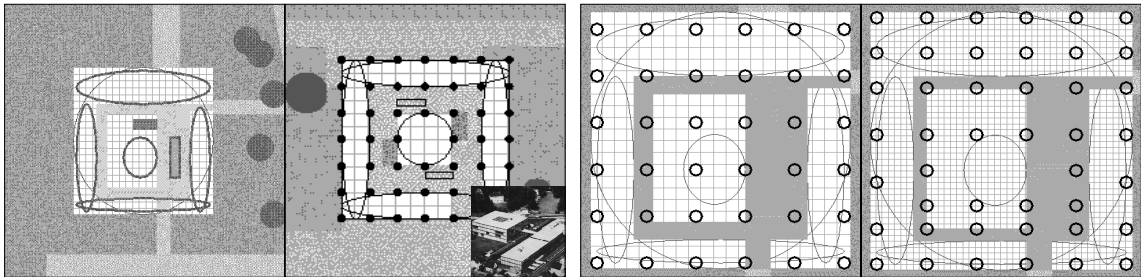


**Bild 40** Wissensrepräsentation, Wissensarten und Problemlösungsmethoden spannen einen Raum von Möglichkeiten auf.

In diesem Kapitel wird gezeigt, daß das Konzept eines vieldimensionalen Datenraumes die etablierten KI-Verfahren aufnehmen kann. Es gibt zur Zeit keine etablierten Klassifikationen von KI-Methoden. Es ist auch nicht das Ziel dieser Arbeit einen Beitrag zu dieser Diskussion zu liefern. Hier wird in Anlehnung an Puppe (Puppe 91) eine Unterscheidung in Wissensrepräsentation, Wissensarten und Problemlösungsmethoden verfolgt. Danach kann nur Wissen formalisiert werden, für das man ein Tripel aus Wissensrepräsentation, Planungsmethode und Wissensart gefunden hat (Bild 40). Dabei lassen sich, wenn überhaupt, nur wenige der vielen möglichen Tripel ausschließen. Wissen, das nicht klar einem Tripel zugeordnet werden kann, ist sehr schwierig zu formalisieren.

## Die Wissensarten

Möchte man die Arbeit im Datenraum durch Automatismen unterstützen, so muß man zunächst unterscheiden, welche Form von Wissen durch die Nutzer einfließen und wie sich diese Wissensarten im Datenraum äußern. Es werden, abgestuft nach dem Grad ihrer Formalisierung, Beispiele fallvergleichenden, probabilistischen, heuristischen und modellbasierten Wissens vorgestellt (Die Definitionen der verschiedenen Wissensarten sind von Puppe (Puppe 91) übernommen worden). Im Bauprozess kann jede dieser Wissensarten identifiziert werden. Beispiele aus MIDI/ARMILLA belegen das.



**Bild 41 a, b, c, d:** Die Bildsequenz zeigt, wie in einer aktuellen Planung ein Problem formuliert werden (das prinzipielle Wege und Nutzungslayout eines Geschosses) und als Anfrage an eine Falldatenbank dienen kann (a). Die Falldatenbank antwortet mit einer oder mehreren vergleichbaren Problemstellungen aus anderen Planungen (b) (episodisches Wissen) oder Katalogen (generisches Wissen). Ein Nutzer oder auch Programme können dort beliebig weitschweifend nach Anregungen für die aktuelle Planung recherchieren (b) (er hat hier die prinzipielle Stützenstellung eingeblendet), z.B. Lösungen oder Lösungswege übernehmen (c) und an die aktuelle Problemstellung anpassen (d). Konnte ein Problem so oder anders erfolgreich gelöst werden, kann es selbst in die Falldatenbank aufgenommen werden und steht dadurch vergleichbaren zukünftigen Problemen zur Verfügung (FABEL 92). Ein gutes Beispiel für CBR in der Architektur: (Domeshek et al. 92).

## Fallvergleichendes Wissen

Fallvergleichendes Wissen ist Wissen über erfolgreich gelöste Fälle, d.h. eine Menge von Problemmerkmalen mit der korrekten Problemlösung. Es wird für Analogie-Schlüsse benutzt, bei denen zu einem neuen Fall ein möglichst ähnlicher, bekannter Fall gesucht wird, um dessen Lösung direkt zu übernehmen oder zu modifizieren (Bild 41).

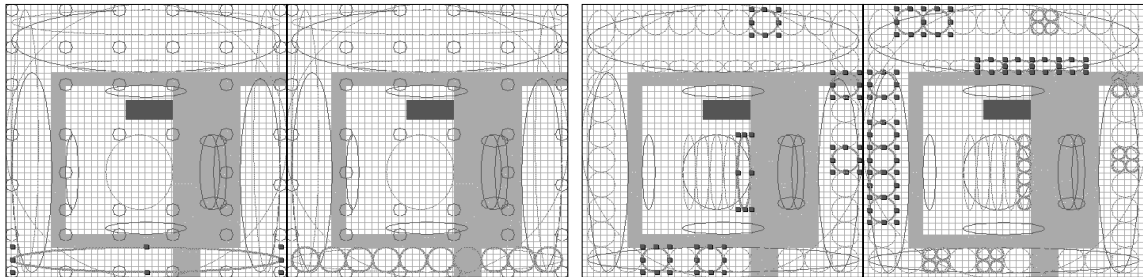
Im Unterschied zum modellbasierten und heuristischen Wissen basiert das fallvergleichende Wissen nicht auf der Grundlage verallgemeinerter Beschreibungen und Verfahren. Hier wird das Wissen eher als Erfahrung denn als Regel aufgefaßt.

Fallvergleichendes Wissen ist zur Zeit Gegenstand der aktuellen Forschung an vielen Stellen, die sich mit computerunterstütztem architektonischem Entwerfen befassen (Oxman 91, Gero 91, Rosenman 91, Shih 91, et al.). Die Diskussionen folgen ungefähr folgender Linie: Während frames und prototypen (modellbasiertes Wissen) allgemeines Wissen anbieten, das von einer bestimmten Situation abstrahiert wurde und an eine neue Situation angepaßt werden muß ("instantiate", "generation", "refinement") und daher operational ist, geht das fallbasierte Wissen von einer ausreichend genauen und deklarativen Beschreibung einer Situation inklusive ihrer Entstehung aus. Aus diesem Grund wird das fallbasierte Wissen mit Geschichte und Gedächtnis in Zusammenhang gebracht, was dem modellbasierten Ansatz abgesprochen wird.

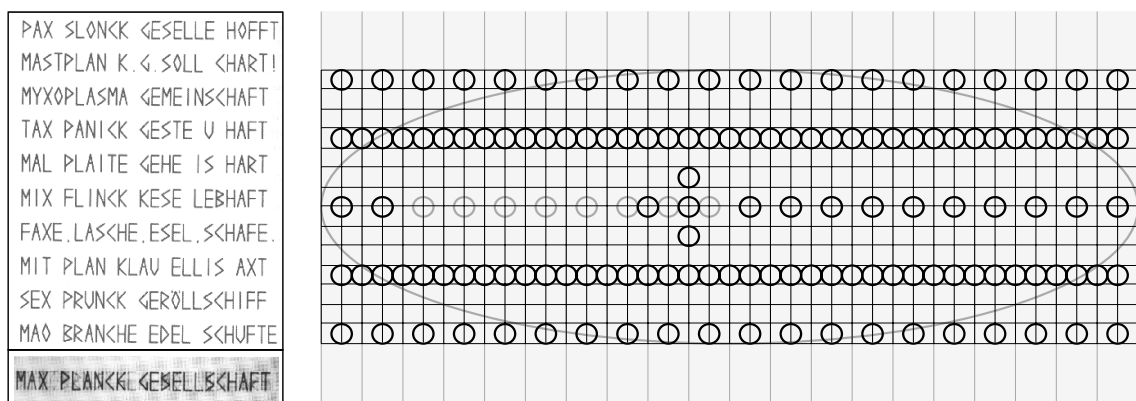
## Heuristisches Wissen

Mit heuristischem Wissen wird oft unsicheres Wissen bezeichnet, das in gerichteter Weise von Problemmerkmalen - eventuell über Zwischenstufen - auf Problemlösungen hindeutet, z.B. Regeln der Art: wenn Problemmerkmal-1, dann Lösungselement-1. Heuristisches Wissen ist zweckgebunden, d.h. leicht interpretierbar, aber selten für multiple Zwecke verwendbar (Bild 42).

Das heuristische Wissen geht davon aus, daß ein Experte über einen reichen Fundus von Beispielen für ein Problem verfügt, die in einem Satz allgemeiner Regeln zusammengefaßt werden können. Deshalb beschreiben Regeln verallgemeinertes Wissen über häufig auftretende Situationen, Probleme und Verfahren. Entspre-



**Bild 42** Diese Bildsequenz zeigt, wie mit heuristischem Wissen auf der Grundlage ungenauer Container neue Container größerer Genauigkeit erzeugt werden können, um so das Design eines Gebäudes schrittweise voranzutreiben (ein unmittelbarer Zusammenhang besteht nur zwischen den Bildpaaren a/b und c/d).

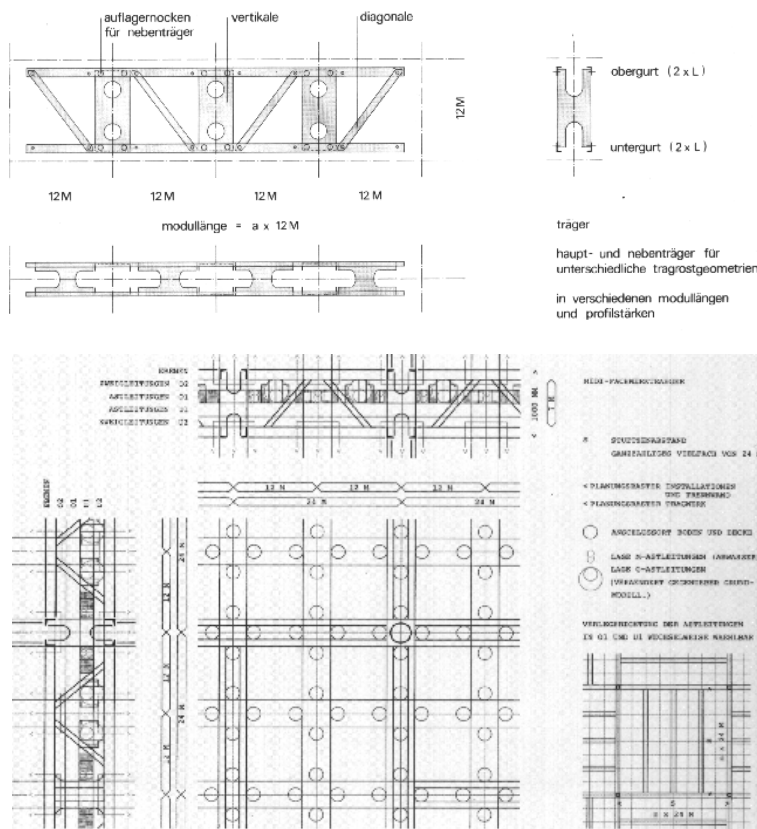


**Bild 43** Bild 41 Das Prinzip der vergleichenden Sequenzanalyse (vgl. (Eigen 87)): individuelle Sätze werden transparent übereinander gelegt und ergeben so einen "typische" Satz (links). Rechts das gleiche Prinzip bei der Anordnung von Abluft-Öffnungen in einem Gebäude: im Hintergrund (grau) die typische Anordnung in fünf regelmäßigen Reihen, im Vordergrund eine individuelle Lösung mit Abweichungen.

chend können heuristische Verfahren genau dann eingesetzt werden, wenn die Situationen eindeutig interpretierbar sind, und wenn alle Entscheidungen auf verallgemeinerbarem Wissen beruhen.

Nach Oxman (Oxman 91) gibt es mit heuristischem Wissen in Zusammenhang mit Designaufgaben verschiedene Probleme, bei denen z.T. andere Wissensarten Vorteile bieten:

- Die Wissensacquisition für Regel-Systeme ist schwierig. Ein menschlicher Nutzer/Planer kann weder jede mögliche Lösung eines Problemkreises identifizieren und beschreiben, noch Regeln formulieren, die allgemein genug wären, um für jede Problemstellung des Problemkreises eine Lösung erzeugen zu können.
- Regel-Systeme haben kein Gedächtnis. Sie können sich nicht an schon gelöste Probleme aber nicht an ihre früheren Fehler erinnern.
- In relevant komplexen Problemkreisen existiert keine direkte Beziehung zwischen einem konkreten Problem und einer allgemeinen Lösung. In dieser Situation müssen sich Experten auf ihre Erfahrung verlassen, um das allgemeine Wissen auf eine konkrete Situation anwenden zu können.



**Bild 44** Beispiele für modellbasiertes Wissen: eine Seite des Kataloges für das MIDI Bausystem (oben) (Haller 74) und eine Seite aus dem ARMILLA Handbuch, einem Regelwerk für den integrierten Entwurf von Leitungssystemen in hochinstallierten Gebäuden (unten) (Haller 85).

## Statistisches (probabilistisches) Wissen

Unsicheres Wissen, das durch statistische Aufbereitung einer Menge erfolgreich gelöster Fälle gewonnen wurde (Bild 43).

## Modellbasiertes (kausales) Wissen

Mit modellbasiertem Wissen werden allgemeine Zusammenhänge zwischen Problemlösungen und Problemmerkmalen bezeichnet, die potentiell in mehrfacher Hinsicht benutzt werden können, aber mit denen eine Problemlösung für die Klassifikation oder Konstruktion meist aufwendiger ist als mit heuristischem Wissen (Bild 44).

In der KI ist diese Wissensart als frames (Minsky 75) oder scripts (Schank 77, Schank 82) bekannt. Nach (Sowa 84) werden mit modellbasiertem Wissen allgemeine und typische Situationen beschrieben. (Gero 87) spricht in diesem Zusammenhang im Bereich des Architektonischen Designs von Prototypen. (Muratori 59) bezeichnet diese Art von Wissen dagegen als Standard (Typen a priori), der bewußt gestaltet ist und grenzt ihn von Prototypen ab, die sich nach seiner Definition entwickelt haben und a posteriori beobachtet werden können.

# Die Problemlöseverfahren

Die oben beschriebenen Wissensarten können für verschiedene Problemlöseverfahren benutzt werden. Puppe (Puppe 90) unterscheidet Klassifikation, Konstruktion und Simulation. Jede dieser Methoden formuliert jeweils nur ein relativ einfaches Verfahren und funktioniert nur in sogenannten Mikrowelten. Diese Mikrowelten zu vergrößern ist bisher nicht gelungen.

Die Umgebungen von Containern können als einzelne Mikrowelten angesehen werden, denn auch sie bestehen in der Regel aus nicht mehr als 100 anderen Containern. Deshalb verfolgt das A4-Modell den Ansatz, die Problemlöseverfahren an einen Container zu koppeln und auf dessen Nachbarn anzuwenden. Für einige Verfahren wird ein einfaches grafisches Schema angegeben, das beschreibt, welchen typischen Einfluß ein Planungsverfahren auf eine Anordnung von Datencontainern in A4 hat.

Die Problemlöseverfahren werden hier recht detailliert beschrieben, obwohl in A4 bisher nur die generate-and-test Strategie genutzt wurde. Es sollen dadurch das Spektrum der Möglichkeiten und implizit auch die Schwierigkeiten einer Integration aufgezeigt werden. Die Problemlöseverfahren im Gebäudeentwurf und -betrieb zu identifizieren, an einigen Stellen im Datenraum zu implementieren und ihre Zusammenarbeit zu integrieren, sollte Gegenstand weiterer Forschungen sein.

## Klassifikation (Diagnostik)



**Bild 45** Klassifikationsmethoden äußern sich ganz allgemein im Datenraum dadurch, daß sie für eine Menge von Containern (Symptome) einen zusammenfassenden Container (Diagnose) suchen. Der Diagnose-Container umschließt die Symptom-Container räumlich.

Die Klassifikation ist ein Problemlösungstyp, bei dem die Lösung, eventuell über einen diagnostischen Mittelbau (= alle Zwischenergebnisse auf dem Weg von Symptomen zu Diagnosen bei der Diagnostik) mit einer Problemlösungsmethode aus einer Menge vorgegebener Lösungen ausgewählt wird. Bei der Diagnostik heißen die Problemmerkmale auch Symptome, die Lösungen auch Diagnosen. Wenn keine Mehrfachdiagnosen zugelassen sind (Single-Fault-Assumption), ist der Lösungsraum bei  $n$  möglichen Diagnosen  $n$ ; mit Mehrfachdiagnosen  $2^n$ .

### Sichere Klassifikation

Klassifikation auf der Basis von sicheren Regeln der Art: Symptom(E) -> Diagnose mit diagnostischem Mittelbau.

### Heuristische Klassifikation

Klassifikation auf der Basis von unsicheren Regeln der Art: Symptom(e) -> Diagnose mit Erfahrungswahrscheinlichkeit  $X$  und diagnostischem Mittelbau. Eine Diagnose wird um so besser bewertet, je höher ihre akkumulierte Erfahrungswahrscheinlichkeit aufgrund der beobachteten Symptome ist.

### Überdeckende Klassifikation

Klassifikation auf der Basis von Regeln der Art: Diagnose(n) (mit bestimmtem Schweregrad) -> Symptom (mit bestimmtem Schweregrad) und diagnostischem Mittelbau. Eine Diagnose wird um so besser bewertet,



je genauer sie die beobachteten Symptome erklären (d.h. überdecken) kann und je weniger nicht beobachtete Symptome sie überdeckt.

#### *Funktionale Klassifikation*

Klassifikation auf der Basis eines Modells des normal funktionierenden Systems, bei dem Eingangsmaterialien mittels Komponenten in Ausgangsmaterialien überführt werden. Symptome sind Diskrepanzen zwischen erwartetem und beobachtetem Wert der (Ausgangs-) Materialien, und Diagnosen sind mögliche Verhaltensänderungen von Komponenten. Wie bei der überdeckenden Klassifikation wird eine Diagnose um so besser bewertet, je genauer sie die beobachteten Diskrepanzen erklären kann.

#### *Fallvergleichende Klassifikation*

Klassifikation auf der Basis einer Sammlung erfolgreich gelöster Fälle. Zu einem neuen Fall wird ein möglichst ähnlicher Fall aus der Fallsammlung gesucht und bei ausreichender Ähnlichkeit dessen Diagnose(n) übernommen.

#### *Statistische (probabilistische) Klassifikation*

Klassifikation auf der Basis einer statistischen Auswertung erfolgreich gelöster Fälle.

#### *Establish-Refine-Strategie*

Problemlösungsmethode für die Klassifikation, bei der in einer strengen Diagnose-Hierarchie zunächst eine Grobdiagnose etabliert und dann verfeinert wird, indem einer ihrer Nachfolger etabliert wird, usw.

#### *Hypothesize-and-Test-Strategie*

Problemlösungsmethode für die Klassifikation, bei der aus den zunächst vorgegebene Symptomen Verdachtsdiagnosen generiert werden, die besten dann gezielt (eventuell auch durch Erfragen zusätzlicher Symptome) überprüft werden und dieser Zyklus solange durchlaufen wird, bis ein Terminierungskriterium gefunden wurde (z.B. daß es keine Verdachtsdiagnosen mehr gibt, oder daß alle Symptome durch etablierte Diagnosen erklärt sind).

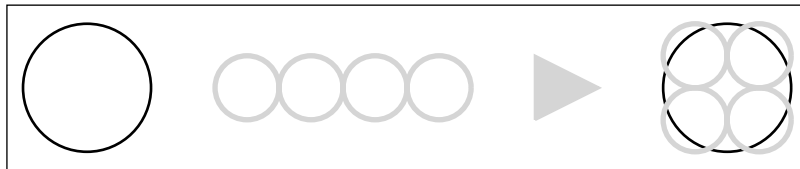
## **Konstruktionsmethoden**

Die Konstruktion ist ein Problemlösungstyp, bei dem die Lösung nicht wie bei der Klassifikation als Ganzes ausgewählt werden kann, sondern mit einer Problemlösungsmethode aus Lösungselementen zusammengesetzt werden muß. Der Lösungsraum ist gewöhnlich außerordentlich groß (z.B. bei einem Stundenplanproblem mit  $n$  Klassen und  $m$  Unterrichtseinheiten pro Klasse in der Größenordnung von  $m^n$ ).

#### *Konfigurierung*

Problemtyp, der zur Konstruktion gehört und bei dem ein Lösungsobjekt gesucht wird, das bestimmte Anforderungen erfüllt. Im Gegensatz dazu ist bei der Planung der Zielzustand (d.h. das Lösungsobjekt) vorgegeben und der Weg dahin gesucht. Im Vergleich zur Zuordnung, bei der die Anordnung vorgegebener Lösungsobjekte gesucht wird, müssen bei der Konfigurierung die Lösungsobjekte instanziiert bzw. ihre Eigenschaften bestimmt werden.

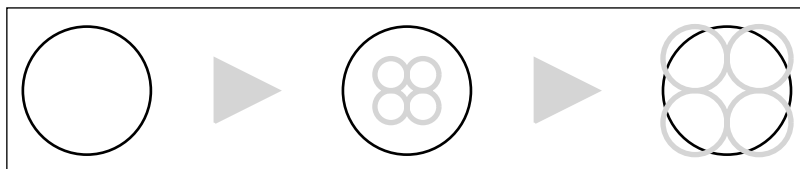
Zuordnung (Bild 46)



**Bild 46** Ein A4-Schema für Zuordnungsprobleme

Problemtyp, der zur Konstruktion gehört und bei dem mindestens zwei disjunkte Mengen von vorgegebenen Objekten so aufeinander abgebildet werden, daß gewisse Anforderungen erfüllt sind. Beispiel: Erstellung von Stundenplänen oder Maschinenbelegungsplänen.

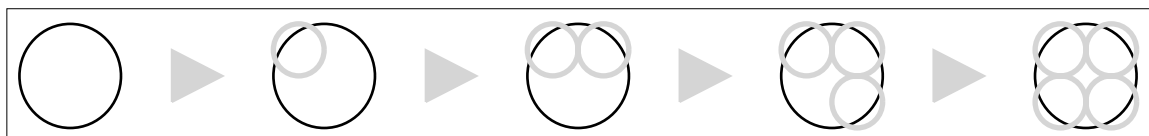
Skelett-Konstruieren (Bild 47)



**Bild 47** Ein A4-Schema für die Skelett-Konstruktion.

Problemlösungsmethode für die Konstruktion, bei der ein Skelettplan aufgrund der Problemmerkmale ausgewählt, expandiert und verfeinert wird. Ein Skelettplan ist eine allgemeine Grobkonstruktion mit einer Menge von abstrakten, zu verfeinernden Lösungselementen. Skelett-Konstruieren ist eine spezielle Form der Generate-and-Test Strategie.

Generate-and-Test Strategie (Bild 48)



**Bild 48** Ein A4-Schema für die Generate-and-Test Strategie.

Problemlösungsmethode für die Konstruktion, bei der die Lösung schrittweise generiert wird, indem ein Lösungselement zur existierenden Teillösung hinzugefügt wird und sofort getestet werden kann, ob die Erweiterung korrekt ist. Im Unterschied zur Hypothesize-and-Test Strategie werden Lösungselemente und nicht Lösungen getestet.

*Vorschlagen-und-Verbessern Strategie (Bild 49)*



**Bild 49** Die prinzipielle A4-Grafik für die Vorschlagen- und Verbessern Strategie

Problemlösungsmethode für die Konstruktion, bei der im Unterschied zur Generate-and-Test Strategie kein sofortiger Test eines Problemlösungsschrittes während der Lösungsgenerierung möglich ist. Falsche Lösungsansätze werden daher eventuell erst spät erkannt und müssen dann verbessert werden. Das Besondere der Strategie im Vergleich zu anderen Backtracking-Ansätzen ist, daß bereichsspezifisches Verbesserungs-Wissen zur effizienten Korrektur falscher Lösungsansätze verfügbar ist.

*Vorschlagen-und-Vertauschen Strategie*

Spezialisierung der Vorschlagen-und-Verbessern-Strategie für Zuordnungsprobleme.

*Least-Commitment-Strategie*

Problemlösungsmethode für die Konstruktion, bei der der Wertebereich der Parameter durch Generierung und Propagierung von Constraints maximal eingeschränkt wird. Dabei wird vermieden, einen Parameterwert, wie bei der Vorschlagen-und-Vertauschen Strategie, ohne zwingenden Grund festzulegen.

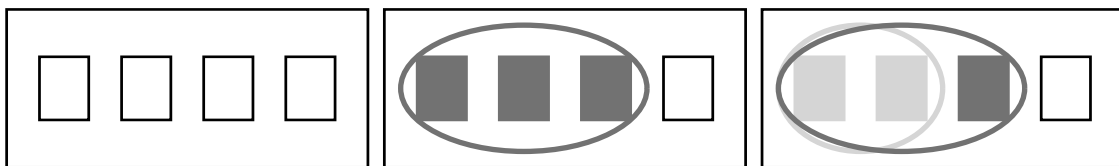
## Simulationsmethoden

Die Simulation ist ein Problemlösungstyp, bei dem die Lösung in der Vorhersage der Auswirkungen von gegebenen Ursachen in einem Modell besteht. Im Gegensatz dazu wird bei der modellbasierten Klassifikation von den gegebenen Auswirkungen (Symptomen) auf vermutete Ursachen (Diagnosen) rückgeschlossen. Man unterscheidet Einphasen-Simulation, bei der für jeden Simulations-Parameter nur ein Wert berechnet wird und Mehrphasen-Simulation, bei der für die Parameter eine Folge von zu verschiedenen Zeitpunkten oder Zeitintervallen geltenden Werten berechnet wird.

# Die Wissensrepräsentationen

Die Wissensrepräsentation ist neben den Wissensarten und den Problemlösungsmethoden die dritte Dimension der Formalisierung von Wissen. Hier sollen vier Wissensrepräsentationsformen vorgestellt werden: Frames, Regeln, Constraints und Truth Maintenance Systems. In A4 gibt es Experimente für die verschiedenen Wissensrepräsentationen. Einige Beispiele belegen dies. In weiterführenden Forschungen müßten sie systematisch im Kontext von A4 untersucht und angewendet werden.

## Frames

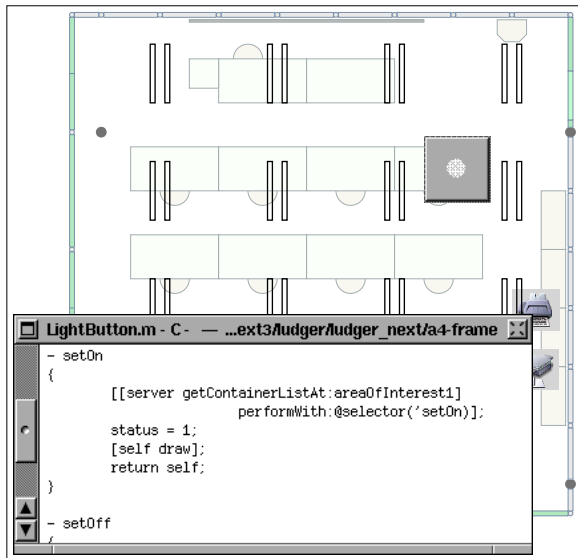


**Bild 50** Container können Attribute-Wert-Paare enthalten. Dieses Bild zeigt die räumliche Vererbung und das Überschreiben. Das Bild links zeigt vier Rechtecke mit der Position 'domain auf der Meta-Achse. Diese Rechtecke haben die Eigenschaft, daß sie sich mit der Farbe darstellen, die sie unter dem Attribut 'farbe an ihrem Ort finden. Im Bild links ist das Attribut 'farbe nicht definiert, d.h. die Rechtecke können keine 'farbe finden und sind transparent. Im mittleren Bild wurde ein weiterer Container installiert, der mit drei der Rechtecke kollidiert und auf der Meta-Achse die Position 'attribute hat. Er enthält den Eintrag 'dark-gray zum Attribut 'farbe. Dadurch sind jetzt die drei linken Rechtecke in der Lage einen Wert 'dark-gray für das Attribut 'farbe zu finden und ändern ihre grafische Representation entsprechend. Im Bild rechts wird ein weiterer Attribut-Container mit der 'farbe 'light-gray installiert. Weil dieser neue Container kleiner, also spezifischer ist als der erste Attribut-Container, finden die beiden linken Rechtecke jetzt zum Attribut 'farbe den Wert 'light-gray und stellen sich entsprechend dar.

Mit Frames wird alles Wissen über ein Objekt in einer Datenstruktur zusammengefaßt. Ein Frame besteht typischerweise aus Verweisen auf übergeordnete Frames, aus Attributen (Slots), Facetten und Werten. Die Facetten können Wissen über Wertebereich, Erwartungswerte (Defaults) oder zugeordnete Prozeduren (Aktionen, die ausgeführt werden sollen, wenn ein Wert eingetragen bzw. abgefragt wurde) enthalten. Die Verweise ermöglichen die Vererbung von Werten und Facetten von dem übergeordneten Frame an seine Nachfolger (Bild 50).

Diese sogenannten objektorientierten Datenstrukturen können wie die Objektstruktur des Designproblems organisiert werden und verhalten sich daher so, wie ein Nutzer es aufgrund der Struktur seines Problems erwartet (Rumbaugh et al. 91). Die objektorientierte Methode hat aber dennoch zwei besondere Nachteile:

- In komplexen objektorientierten Systemen muß oft Wissen repräsentiert werden, das einen Zusammenhang mehrerer gleichrangiger Objekte bezeichnet und keinem der Objekte klar zugeordnet werden kann. Regeln sind eine mögliche Antwort auf dieses Problem.
- Objektorientierte Systeme gehen implizit davon aus, daß auf der Grundlage sehr vieler Einzelfälle eindeutig allgemeine Klassen beschrieben werden können, die das Verhalten ihrer Instanzen (die Beschreibung der Einzelfälle) festlegen. Für viele Probleme gibt es aber nicht die Erfahrung vieler vergleichbarer Einzelfälle, so daß - wenn objektorientierte Strukturen überhaupt eine adäquate Methode sein sollten - die Klassen spekulativ und damit für Fremde wenig verständlich werden. Untersuchungen um case-based reasoning sind eine mögliche Antwort auf dieses Problem.



**Bild 51** Ein Beispiel für Regeln in A4. Die Leuchten eines Seminarraumes sollen mit einem Schalter ein- und ausgeschaltet werden können. Dazu bekommen die Leuchten und der Schalter die gleiche Position auf der Kompositions-Achse. Der Lichtschalter kennt die Methoden 'setOn' und 'setOff'. Der Objective-C Code (Fenster unten links) beschreibt, welche Regeln ablaufen, wenn der Schalter eingeschaltet wird: zunächst wird eine Liste von Nachbar-Containern auf der Grundlage einer 'areaOfInterest' von einem Server erfragt. In diesem Fall ist das Kriterium für die Nachbarschaft die gemeinsame Position auf der Kompositions-Achse. Der Server antwortet mit einer Liste aller Leuchten des Raumes. Dann wird jeder dieser Leuchten die Methode 'setOn' geschickt, der eigene Status auf 1 (= an) gesetzt und der Schalter im eingeschalteten Zustand neu gezeichnet.

## Regeln

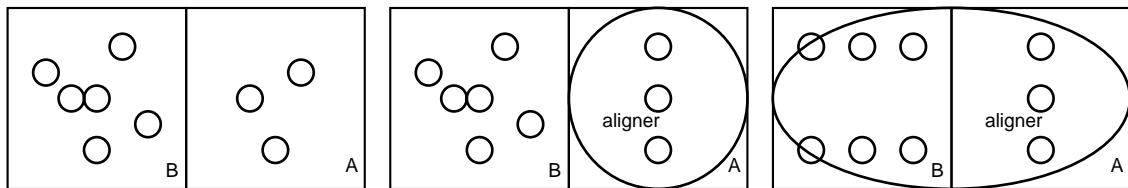
Mit Regeln werden gerichtete Beziehungen zwischen Objekten dargestellt. Im Gegensatz zu Regeln werden ungerichtete Beziehungen zwischen Objekten mit Constraints dargestellt. Eine Regel (z.B. A -> B) hat immer eine Vorbedingung und eine Aktion, welche dann ausgeführt werden darf, wenn die Vorbedingung erfüllt ist. Die beiden grundlegenden Ableitungsstrategien für Regelsysteme sind Vorwärts- und Rückwärtsverkettung. Bei der Vorwärtsverkettung werden die Regeln danach ausgewählt, ob ihre Vorbedingung durch die Daten in der Datenbasis erfüllt ist, und bei der Rückwärtsverkettung danach, ob ihre Aktion zur Herleitung eines vorgegebenen Ziels beiträgt.

Eine gute Übersicht über die Möglichkeiten von Regelsystemen in Zusammenhang mit Design wird in (Coyne 88) gegeben. (Tomiyaama 87b) unterscheidet extensionale (Regeln) von intensionalen (Frames) Beschreibungen. Nach dieser Unterscheidung gehen die Frames von vordefinierten Bedingungen mit großer Performance und geringer Anpassungsfähigkeit aus, während die Regeln keine besonderen Annahmen treffen und konzeptionell sehr anpassungsfähig sind, aber eine im Vergleich schlechte Performance bieten. CAD Systeme brauchen hohe Performance und großes Flexibilität. (Bijl 89) propagiert grundlegend neue, sehr anpassungsfähige CAD Systeme auf der Basis von Regeln als Alternative zu den üblichen modernen Systemen auf der Basis von Frames (Bild 51).

## Constraints

Mit Constraints (Bild 52) werden ungerichtete Beziehungen zwischen Objekten ausgedrückt. Ein Constraint besteht aus Name, Definition und Variablen (z.B. "Addierer-1", "A + B = C", (A, B, C)). Constraints, die durch gemeinsame Variablen verbunden sind, bilden ein Constraint-Netz. Ein Constraint-Netz mit einer Anfangsbelegung einer Teilmenge der Variablen heißt Constraint-Problem.

Auf das Design übertragen könnte z.B. ein auf Constraints basierendes Designmodell etwa so aussehen: Konstruieren ist ein Prozeß des Suchens nach einer Lösung aus einer Menge möglicher Lösungen. Die gesuchte Lösung muß bestimmten Kriterien (Constraints) genügen. Ein Lösungszustand ist erreicht, wenn



**Bild 52** Die Wirkungsweise von Containern, die mit Constraints ausgestattet sind. Im Bild links haben die rechteckig dargestellten Container A und B ein mittels Constraints formuliertes Interesse an einer spezifischen Anzahl Kreisen innerhalb ihres jeweiligen Bereiches. Innerhalb des Bereiches des Containers A wird es daher z.B. ohne Protest nicht möglich sein, einen Kreis zu löschen oder hinzuzufügen. Wohl aber ist es möglich die Kreise innerhalb des Containers A z.B. neu anzuordnen. Dies kann entweder durch einen Nutzer, oder durch einen weiteren Container geschehen, der hier 'aligner' genannt werden soll (Mitte). Er ist mittels Constraints mit dem Interesse ausgestattet alle Kreise innerhalb seines Bereiches regelmäßig anzuordnen. Das Bild in der Mitte zeigt, wie der Container Aligner in dem Bereich des Containers A angeordnet wurde. Die drei Kreise stehen jetzt im Wirkungsbereich der Constraints von A und dem Aligner. Deshalb kann es in diesem Bereich nur genau drei Kreise (= Interesse von A) in regelmäßiger Anordnung (= Interesse des Aligners) geben. Das Bild rechts zeigt was passiert, wenn der Interessenbereich des Aligners auch auf den Bereich von Container B ausgedehnt wird.

diese Kriterien erfüllt sind. Ein Lösungszustand kann damit mehrere Lösungen umfassen (Kalay 87). Beispiele sind: LOOS (Flemming 78 89), MOLGEN (Stefik 81) oder GARI (Descotte 85).

## Truth Maintenance System

Als Truth Maintenance System werden Techniken der konsistenzhaltenden Rücknahme von Schlußfolgerungen aufgrund einer geänderten Situation (z.B. Verbesserung des Faktes durch den Benutzer, Korrektur einer Default-Annahme aufgrund neuer Informationen) bezeichnet. Einfache, aber ineffiziente Techniken sind die vollständige Neuberechnung aller Schlußfolgerungen auf der Basis der neuesten Daten oder chronologisches Backtracking. Effizientere Techniken verwenden eine zusätzliche Datenstruktur zum Abspeichern von Begründungen. Beim JTMS (justification-based TMS) werden direkte Begründungen einer Schlußfolgerung abgespeichert; beim ATMS (Assumption-Based TMS) zusätzlich ihre zugrundeliegenden Basisannahmen.

TMS haben große Relevanz für die Implementation von Designmodellen, die von einem nicht-linearen Vorgehen ausgehen. Beispiele sind MOLGEN (Stefik 81) oder IICAD (Veth 87).

# 6 Datenbanken und das A4-Modell

*Nachdem die verschiedenen Möglichkeiten für Automatismen innerhalb der einzelnen Container vorgestellt worden sind, geht es in diesem Abschnitt um die Koordination der verschiedenen Wissensbestände. Das ist die Domäne von Datenbanksystemen und Blackboard Systemen. Sie haben aber verschiedene Nachteile. Für A4 bietet sich als Alternative eine räumliche Koordination der Wissensbestände an. Dieses neue Koordinationsprinzip müßte durch weiterführende Forschungen mit einem Daten- und Transaktionsmanagement auf Datenbankniveau ausgestattet werden. Dabei hilft die für A4 besonders wichtige grafische Koordination.*

## Passive Objekte

(Ahmed 90) beschreibt neun Vorteile der neueren objektorientierten Datenbanken (Atkinson 89) gegenüber den relationalen Datenbanken (Katz 85: CAD und Datenbanken), (Mattos 90: relationale Datenbanken und KI-Techniken), (Karagiannis 90: DBMS und KI):

### *Ein realistischeres Datenmodell*

Die Klassen und Objekte der OODBMS repräsentieren real-world Einheiten und entsprechen daher der Struktur eines Problems weit besser als die flachen tables der RDBMS.

### *Ein leistungsfähigeres Datenmodell*

Durch Methoden wie inheritance und schema evolution sind OODBMS relativ flexibel und ermöglichen eine Anpassung der Datenbank an die spezifischen Erfordernisse eines Nutzers mit wenig Restriktionen. RDBMS sind erheblich restriktiver und daher starrer.

### *Einfacherer Entwurf des Datenmodells*

Generalization und inheritance ermöglichen eine bessere und intuitivere Strukturierung der OODBMS, die dazu führt, daß das Datenmodell kompakter und übersichtlicher gestaltet werden kann.

### *Independence mismatch in RDBMS*

Die Sprache zur Manipulation von RDBMS (z.B. SQL) hat in der Regel eine andere Struktur als die allgemeinen Programmiersprachen, die die Datenbank nutzen. Dadurch kommt es zu Konflikten. OODBMS können dagegen vollständige Programmiersprachen integrieren (z.B. C++, CLOS oder Smalltalk).

### *Identität der Objekte*

RDBMS sind wertebasiert, d.h. Relationen zwischen Objekten basieren auf gleichen Werten ihrer entities.

Um Objekte voneinander zu differenzieren muß ein Nutzer sehr viele eindeutige entities etablieren, wodurch Redundanzen und unübersichtliche Situationen entstehen. Objekte in OODBMS bekommen dagegen unabhängig von ihren Attributen einen eindeutigen Namen.

#### *Leistungsfähigere und einheitliche Wissensrepräsentation*

Objekt-orientierte Datenrepräsentation integriert Daten (Attribute) und Wissen (Prozeduren). Ein Reihe von KI-Techniken werden in OODBMS angeboten, wie Klassifikation, Spezialisierung in Hierarchien, Delegation von Verhalten, Polymorphismus und run-time binding von Nachrichten und Methoden, automatische garbage collection und Speicherverwaltung.

#### *Besseres transaction management und concurrency*

OODBMS kennen das Verhalten ihrer Objekte und können daher concurrency und transactions besser gestalten als der serielle no-read-write Ansatz der RDBMS.

#### *Eine leistungsfähigere Anfragesprache*

Die Anfragesprache von OODBMS kann, weil sie allgemeine Programmiersprachen integriert, an spezielle Probleme angepaßt werden. Es fehlt aber eine standardisierte Anfragesprache wie z.B. SQL.

#### *Bessere Unterstützung von kooperativem Arbeiten*

OODBMS können die kooperative Zusammenarbeit besser unterstützen als RDBMS, weil sie die Verwaltung komplexerer Daten ermöglichen, das gleichzeitige Nutzen von Dateneinheiten verschiedener Größen, sowie Versionen und Alternativen unterstützen und besser auf Reihenfolgen und Abhängigkeiten von Entscheidungen eingehen können.

Trotz dieser Leistungsfähigkeit von OODBMS haben sie, wie alle objekt-orientierten Systeme, für das A4-Modell einige Nachteile:

#### *Die Objekt-Orientierung*

Objekt-orientierte Modelle sind ein adäquates Mittel für die Repräsentation der Struktur und das Verhalten von isolierten Artefakten. Sie haben jedoch Schwierigkeiten mit der Beschreibung der komplexen Abhängigkeiten dieser Artefakte in Planungsumgebungen.

#### *Die zentrale Koordinierung*

OODBMS benötigen eine zentrale Koordinierung, um z.B. die Kollisionen mehrerer Nutzer zu verhindern. Das ist bei Designaufgaben, mit ihren sehr unvorhersehbaren und verstreuten Datenbankzugriffen und mit ihren langen Transaktionen, ein erhebliches Problem und endet in der Praxis oft mit nicht nachvollziehbaren Reaktionen. Gleichzeitig ist die zentrale Koordinierung ein funktionaler, struktureller und operationaler Engpaß.

#### *Der Verwaltungsaufwand*

Besonders in großen objekt-orientierten Systemen kommt man wegen der geforderten homogenen Struktur schnell in die Situation, daß man, nur um eine kleine zusätzliche Funktion hinzuzufügen, ein Objekt mit sehr vielen Attributen und Methoden kennen, spezifizieren und bearbeiten muß, um sie richtig in die globale objekt-orientierte Struktur einzuordnen. Dieser Aufwand steht oft in keinem Verhältnis zu z.B. einer funktionalen Beschreibung.

#### *Die Vorstrukturierung*

Objekt-orientierte Systeme gehen immer davon aus, daß aus vielen Einzelfällen Klassen abstrahiert werden



können, die das Verhalten der späteren Instanzen festlegen. Gerade im Bereich der Designprobleme gibt es aber oft nur Einzelfälle, die mangels Erfahrung nur schwer strukturierbar sind. Eine Klassifikation ist daher in der Praxis oft willkürlich, kann schwer von anderen Personen nachvollzogen werden und ist daher nicht das geeignete Mittel.

## Aktive Objekte

Es gibt im Zusammenhang mit Datenbanken Forschungen, die über die Möglichkeiten von OODBMS hinausgehen und für das A4-Modell geeigneter erscheinen. Prominent sind die ACTOR Modelle, die in (Agha 86) zusammengefaßt sind. Sie bezeichnen selbständige, sogenannte Aktive Objekte (diese würden den mit Wissen ausgestatteten Containern entsprechen), die sich aus eigenem Antrieb sozial organisieren. Sie stehen damit im Gegensatz zu den oben beschriebenen passiven Objekten. Passive Objekte können zwar auch Aktionen ausführen, benötigen dafür aber immer einen äußeren Anstoß und üblicherweise eine zentrale Kontrolle.

Ein sehr weitreichendes Beispiel für ein ACTOR Modell ist das GOM Projekt (Kemper 90). Es spezifiziert seine Aktiven Objekte auf eine sehr verständliche Art:

Objekte sind sinnvoll nicht weiter zerlegbare Einheiten, die ihre interne Struktur von ihrer externen Umgebung abgrenzen. Objekte müssen einerseits ihren internen Zustand speichern können und andererseits in der Lage sein, mit anderen Objekten zu kommunizieren. Sie bestehen daher aus drei Komponenten:

### *structural body*

Der structural body speichert den internen Zustand eines Objektes und beinhaltet mindestens den eindeutigen Namen des Objektes

### *behavioral knowledge*

Jedes Objekt reagiert auf eine spezifische Art auf externe Ereignisse.

### *sensory system*

Sensoren beschreiben, wie ein Objekt seine Umgebung "sehen" kann.

Die so ausgestatteten Objekte können nach den bekannten Methoden klassifiziert und hierarchisiert werden, es können Prototyen zum Kopieren gebildet werden. Diese Objekte können auch für komplexere Aufgaben zusammenarbeiten. Es sollen einige Beispiele für eine Szenerie einer solchen Zusammenarbeit gegeben werden:

### *sites and governments*

Die Welt der aktiven Objekte ist auf sehr viele Rechnerumgebungen (sites) verteilt. Jede dieser Rechnerumgebungen hat - entsprechend autonomen Ländern - eine Regierung, die ihre Bewohner (Autonome Objekte) und Besucher lenkt. Darüber hinaus ist die Regierung für die grundlegende Infrastruktur innerhalb der Rechnerumgebung verantwortlich, die von den Objekten benutzt werden kann.

### *type manager*

Der type manager ist verantwortlich für die Taxonomien einer site, die sich dynamisch ändern müssen, weil

die Objekte reisen können.

#### *intra- and inter-site communication channels*

Objekte und Nachrichten können zwischen den sites verschickt werden. Eine site bietet den Objekten und Nachrichten spezifische Möglichkeiten: Manche sites sind darauf spezialisiert Besucher zu empfangen wie z.B. Hotels, Agenturen oder Schulen. Andere akzeptieren dagegen nur Objekte und Nachrichten von bestimmten sites und bestimmtem Typ.

#### *mailboxes*

Weil die Objekte mobil sind, bieten mailboxes die Möglichkeit Nachrichten zu hinterlegen.

#### *methamorphosis*

Die Objekte können im Verlauf ihrer Lebenszeit ihre interne Struktur und ihr Verhalten ändern. Besonders ändern sie ihren Ort im System und damit ihre Nachbarschaften.

#### *object communication pattern*

In einer verteilten Umgebung für Autonome Objekte spielt die Kommunikation eine entscheidende Rolle. Auf der grundlegenden Ebene können synchrone von asynchronen Nachrichten unterschieden werden. Auf einer höheren Ebene können Konferenzen für eine enge Zusammenarbeit verschiedener Objekte veranstaltet werden, es können beeper installiert werden, die ein Objekt rufen, wo immer es sich befindet, oder es gibt Sekretärinnen, die in Abwesenheit eines Objektes Kommunikationsaufgaben übernehmen.

GOM, wie andere Forschungen um Aktive Objekte (e.g. (Caromel 88), (Dayal et al. 88), (Rao et al. 89), (Chamberlin 90), (Delattre et al. 90), (Saiedian et al. 90)) befinden sich auf der Ebene der Grundlagenforschung und zeigen noch keine praktisch oder sogar kommerziell nutzbaren Ergebnisse. Eine Implementation des A4-Modells muß daher auf die etablierten Prinzipien passiver Objekte zurückgreifen, benutzt aber die Aktiven Objekte als eine dem A4-Modell entsprechende Metapher.

## WAVE

Ein exotischer Ansatz für die Verwaltung eines semantischen Netzes, der verteiltes und paralleles Arbeiten erlaubt, ist das WAVE System von (Sapaty 88, 91). Es besteht aus einer theoretisch unbegrenzten Anzahl von Knoten, die mit anderen Knoten verbunden sind. Jeder Knoten verfügt über einen WAVE Interpreter. In diesem Netz können an verschiedenen Stellen gleichzeitig die sehr kompakten WAVE Programme injiziert werden. Das WAVE System ist nun nicht so organisiert, daß wie z.B. in objekt-orientierten Systemen, Daten von einem Objekt zu einem anderen Objekt verschickt werden. Das injizierte Programm wird vielmehr von dem lokalen Knoten interpretiert, u.U. modifiziert und an seine Nachbarknoten weitergeschickt, die es ihrerseits - parallel ! - interpretieren, modifizieren und weiterschicken. Die injizierten WAVE Programme breiten sich wie Wellen in dem semantischen WAVE-Netz aus, können sich überlagern, blockieren sich nicht gegenseitig und können vollständig verteilt und parallel in einem physikalisch heterogenen Netz bearbeitet werden.

Interessant ist, daß die WAVE Programme sehr kompakt sind, daß sich die Welle unabhängig vom augenblicklichen Zustand des Netzes ausbreitet und daß die WAVE Programme bildlich - und nicht, wie bisher üblich, sequentiell - organisiert werden können. Der erste Schritt zu einem WAVE Programm ist tatsächlich eine bildliche Vorstellung von dem, was im dem semantischen Netz passieren soll. Das Ergebnis auf diese

bildliche Anfrage kann mit einer Reflexion der Anfrage durch das konkrete semantische Netz beschrieben werden. Dabei sind die Bilder verschiedener Injektionen unterschiedlich eingefärbt und heben sich selbst bei Überlagerungen deutlich voneinander ab. Gleichzeitig ist der WAVE Interpreter so kompakt, daß es möglich scheint, einen WAVE Prozessor als Basis für einen hochgradig parallel arbeitenden WAVE Computer zu entwickeln. Zur Zeit ist WAVE in LISP implementiert und kann sich über TCP/IP in einem LAN ausbreiten.

Für das A4-Modell ist WAVE wegen seiner strukturellen Parallelisierung und Verteilung neben der bildlichen Programmierung interessant. Es liegen aber noch keine besonderen Erfahrungen mit einer Kopplung beider Systeme vor. Vor allen Dingen müßte die Vorstellung selbstständiger, kooperierender, aktiver Objekte, die diese Arbeit durchzieht, revidiert werden, so daß in dieser Arbeit keine weiteren Bezüge zu WAVE aufgebaut werden.

## Blackboard Systeme

In Bereich der künstlichen Intelligenz sind Blackboard Systeme die prominentesten Koordinatoren verschiedener Wissensbestände (Engelmore 88). Sie organisieren die Kombination von verschiedenen KI-Techniken nicht durch eine homogene, mächtige und die verschiedenen KI-Techniken umfassende Sprache (wie z.B. in (Puppe 90)), sondern durch ein Kommunikationsmodell, an dem sich im Prinzip beliebige KI-Techniken beteiligen können (Bild 53). Diese werden in sogenannten Knowledge Sources (KS) repräsentiert und konzentrieren sich auf einzelne Spezialaufgaben. Eine neutrale Beschreibung des Gesamtproblems liegt auf dem sogenannten Blackboard (BB). Es kann von allen KS des Systems gelesen und bearbeitet werden und dient somit als Medium für die Zusammenarbeit der KS. Der Arbeitsablauf funktioniert im Prinzip so: eine KS bekommt von einer zentralen Steuerung (die selbst eine KS sein kann) den Auftrag, ihre Teilplanung durchzuführen. Die Ergebnisse werden auf das BB geschrieben und sind damit potentiell für alle anderen KS verfügbar. Danach folgt eine Analysephase, die auch von KS übernommen werden kann. Sie ermittelt, welche KS sinnvollerweise als nächstes ihr Wissen in die Planung einbringen kann. Sie bekommt von der Steuerung den Auftrag, diese Teilplanung durchzuführen. Die Ergebnisse werden erneut auf das BB geschrieben usw.

Es gibt eine ganze Reihe von BB-Systemen, die sich mit Spracherkennung, Bilderkennung, Planung, Softwareentwicklung oder Real-Time AI befassen. In (Engelmore 88) gibt es ein umfangreiches Diagramm der historischen Entwicklungen der verschiedenen BB-Modelle. Sehr prominent ist die Programmierumgebung für BB-Systeme BB1 (Hayes-Roth 84) und - für die architektonische, räumliche Planung interessant - das in BB1 implementierte System ACCORD (Hayes-Roth 86), mit dem räumlich-semantische Zusammenhänge formuliert werden können. Neuere Kopplungen von BB-Systemen mit CAD im Bereich Architektur finden

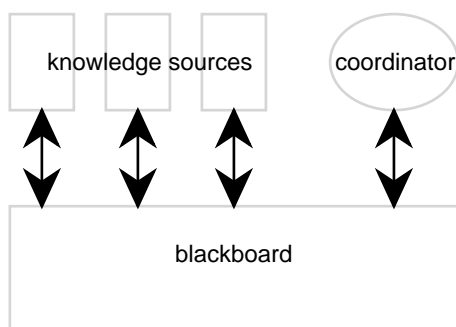


Bild 53 Typischer Aufbau eines Blackboard-Systems.

sich z.B. in ICADS (Pohl 88) oder AEDOT (Stratton 90).

Das A4-Modell unterscheidet sich in drei Bereichen von Blackboard Systemen:

#### *Deklaration*

Dem Autor ist kein Blackboard System bekannt, in dem die Interaktion eines oder mehrerer Nutzer mit dem Blackboard System eine konstituierende Rolle spielen würde. Sie gehen alle von der Annahme autonomer Automaten aus, die Experten ersetzen oder wenigstens möglichst vollständig simulieren sollen. Die Möglichkeit einer Interaktion von Nutzern würde die Blackboard Architektur auch erheblich verändern, denn sie würde die Rolle der zentralen koordinierenden Steuerung relativieren. Die Knowledge Sources, von denen manche durch Nutzer gesteuert würden, würden asynchron und partiel autonom arbeiten. Es gäbe unvorhersehbare Konflikte. Es entstünden Fragen der Transparenz der Blackboard Datenstrukturen für die Nutzer usw. Systeme wie ICADS (Pohl 88), die CAD mit BB zu verknüpfen versuchen, d.h. Nutzereingaben in ein BB-System erlauben, arbeiten mit einer strikten Trennung beider Systeme, so daß nur entweder ein Nutzer oder das BB-System arbeiten können.

#### *Vorstrukturierung*

Bei aller Flexibilität von Blackboard Systemen sind sie doch zentralisiert und vorstrukturiert. Es gibt ein zentrales Blackboard, es gibt eine zentrale Steuerung und es gibt eine Menge von programmierten Knowledge Sources. Diese komplexen Strukturen müssen, wenn sie leistungsfähig sein sollen, auch spezialisiert sein, so daß es ein allgemeines und gleichzeitig kompetentes System nicht geben kann.

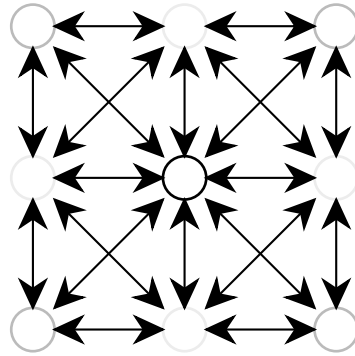
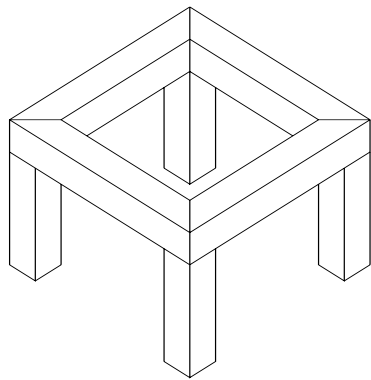
#### *Verteilung*

Es gibt zwar Blackboard Systeme, v.a. für Echtzeitprobleme, die mit verteilten oder hierarchischen Blackboards arbeiten, und die auch das parallele und asynchrone Arbeiten der Knowledge Sources ermöglichen (Engelmore 88). Aber auch hier - obwohl es sich anbieten würde - ist 1) die menschliche Interaktion nicht konstituierend und 2) die verteilte Blackboard Struktur vordefiniert, damit eine vollständige zentrale und automatische Kontrolle der Knowledge Sources untereinander jederzeit gewährleistet ist. Dies entspricht auch der Aufgabe dieser Systeme v.a. im militärischen Überwachungsbereich.

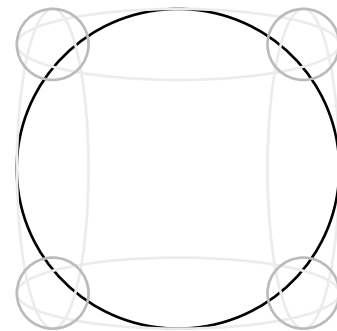
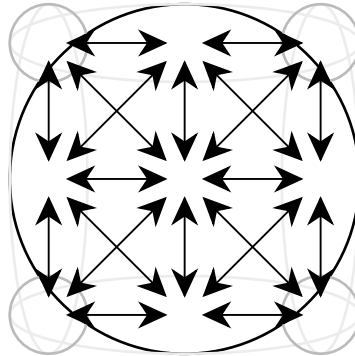
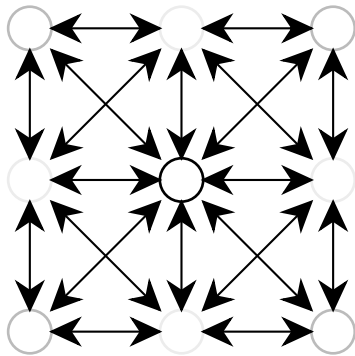
## Die Koordination im A4-Modell

A4 verfolgt in Anlehnung an die Diskussion über aktive Objekte einen zu den Blackboard Systemen alternativen Ansatz, der mit der räumlichen Position der Container in direktem Zusammenhang steht (Hovestadt 91) (Gauchel 92a) (Gauchel 92b) (Bhat 93):

Als einfaches Beispiel kann ein minimales MIDI Tragwerk dienen (Bild 54a). Es besteht aus vier Stützen und vier Trägern. Bild 54b zeigt die semantischen Abhängigkeiten dieser Bauteile untereinander: die dunkelgrauen Knoten repräsentieren die Stützen, die hellgrauen Knoten die Träger und der zentrale schwarze Knoten auf einer abstrakteren Ebene das Gesamtsystem. Die radial dargestellten Relationen bezeichnen die Relation 'part-of' vom Gesamtsystem zu seinen Komponenten (Träger und Stützen). Die konzentrisch angeordneten Relationen bezeichnen die Relation 'konstruktiv-verbunden-mit' der Träger und Stützen untereinander.



**Bild 54 a, b:** Links ein minimales MIDI Tragwerk, rechts ein entsprechender Graph. Die hellgrauen Knoten entsprechen den Trägern, die dunkelgrauen Knoten den Stützen, der zentrale schwarze Knoten dem Gesamtsystem.

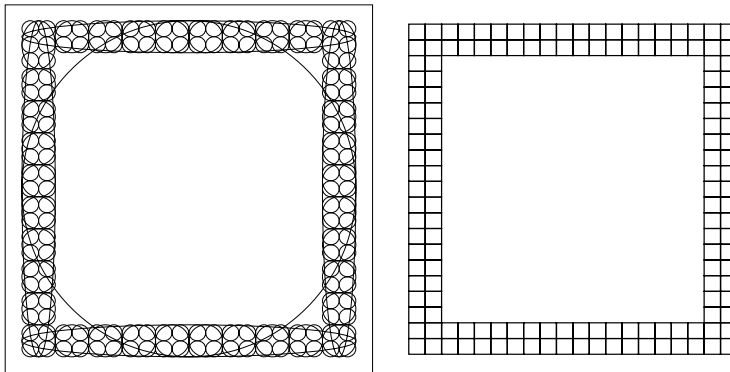


**Bild 55** Die Überführung üblicher Graphendarstellungen semantischer Netze in eine entsprechende A4-Darstellung. Links der schon in Bild 54 beschriebene Graph. In der Mitte ist der gleiche Graph dargestellt, nur sind die Knoten jetzt räumlich angeordnet. Ein wichtiger Effekt ist, daß die Relationen in der Regel aus der räumlichen Überlagerung der Knoten gelesen werden können und somit auf eine grafische Repräsentation durch Pfeile verzichtet werden kann (Abbildung rechts).

Diese Abhängigkeiten der Daten werden üblicherweise direkt in Datenbanken oder Blackboard Systemen abgelegt. Der in Bild 54b dargestellte Graph zeigt symptomatisch die besonderen Schwierigkeiten dieser Repräsentation: zwar sind die semantischen Abhängigkeiten sehr präzise dargestellt, die räumlichen Abhängigkeiten dagegen überhaupt nicht. Zudem ist dieser Graph sehr schwer von einem Nutzer interpretierbar, weil die Knoten trotz unterschiedlicher Bedeutung (in diesem Beispiel: Träger, Stützen und Gesamtsystem) gleich dargestellt sind. Außerdem benötigt ein Graph dieser Art sehr viel Darstellungsfläche pro Informatonseinheit, sodaß nur kleine Graphen angezeigt werden können.

Das A4-Modell ergänzt die semantischen Strukturierungsmöglichkeiten durch räumliche Strukturen und erreicht dadurch leistungsfähigere Datenstrukturen, die sich zudem verständlicher darstellen und editieren lassen. Im A4-Modell gilt die Regel, daß zwei Knoten (sie entsprechen den Containern des A4-Modells) nur dann Relationen zueinander aufbauen können, wenn sie im vieldimensionalen Datenraum räumlich benachbart sind (vgl. Tabelle 1). Die räumlichen Nachbarschaften der Stützen und Träger im Beispiel aus Bild 54 ergeben direkt den Graphen der Relation 'konstruktiv-verbunden-mit'. Etwas schwieriger ist die räumliche Entsprechung der Relation 'part-of zwischen dem Knoten 'Gesamtsystem' und den Knoten seiner Komponenten. Damit diese Relation im A4-Modell verwaltet werden kann, muß der Knoten 'Gesamtsystem' die Knoten seiner Komponenten räumlich umfassen.

So wie in A4 die semantischen Netze räumlich überformt werden, werden auch die Graphendarstellungen



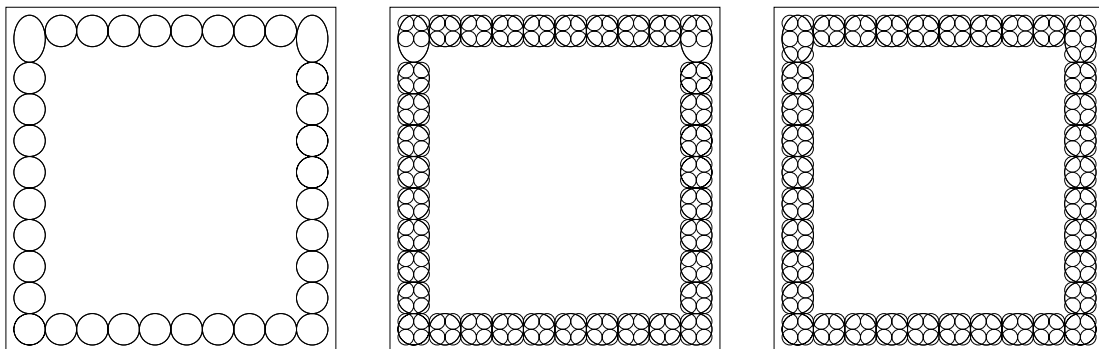
**Bild 56** A4 bevorzugt Grafikelemente, die sich gegenseitig nur wenig überlagern. Das sind z.B. Kreise, Ellipsen, Kurven oder Texte im Gegensatz zu Rechtecken, Polylines u.a. Diese Abbildungen zeigen die besonderen Möglichkeiten einer Ellipsendarstellung von Knoten bzw. Containern. Links sind einander überlagernde Knoten aus vier verschiedenen Hierarchiestufen gleichzeitig dargestellt. Das Bild zeigt einen Baum mit vier Hierarchieebenen und ca. 200 Knoten, deren relative Lage zueinander gut beobachtet und editiert werden kann. Mit einer Darstellung der Knoten durch Rechtecke (rechts) können die Hierarchieebenen nur einzeln dargestellt und daher nicht aufeinander abgestimmt werden.

räumlich überformt: die Knoten des semantischen Netzes werden entsprechend ihrer räumlichen Bereiche dargestellt, die Relationen erschließen sich durch die räumliche Überlagerung der Knoten und brauchen in der Regel nicht dargestellt werden (Bild 55). Das Ergebnis ist deutlich übersichtlicher (die Knoten können z.B. mit CAD-Methoden entsprechend ihrer Bedeutung dargestellt und editiert werden) und kompakter (Bild 56).

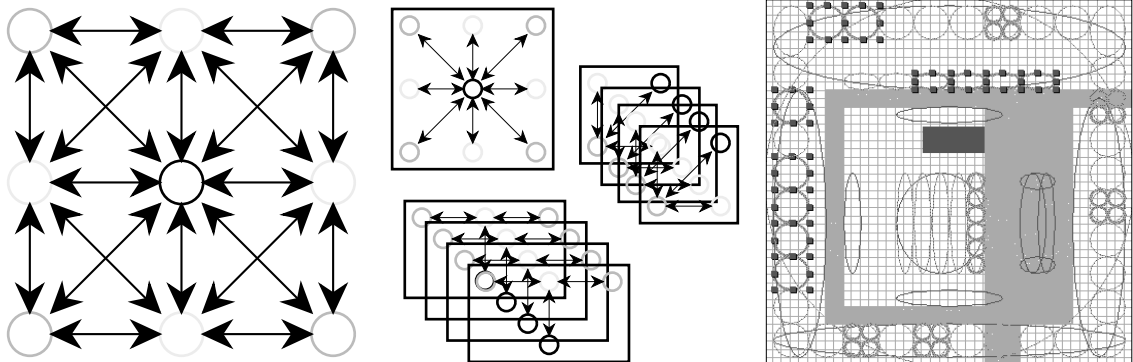
#### Die Auswirkungen auf die Wissensmodellierung

Die durch diese Strukturierungs- und Darstellungstechnik erreichte gute Übersicht über komplexe Zusammenhänge für einen Nutzer entschärft die Anforderungen an die Wissensmodellierung und macht v.a. deren Anwendung deutlich anpassungsfähiger. Ein Beispiel soll das verdeutlichen (Bild 57):

Im Verlaufe des Entwurfes einer Klimaanlage für ein Bürogebäude wurden 36 Container für Zuluft-Mischgeräte entlang der Fassade und im Deckenhohlraum des Gebäudes installiert (Bild 57a). Der nächste Planungsschritt ist die Anordnung der Zuluft-Auslässe. Da in der Regel jedes Mischgerät 4 Auslässe versorgt, kann jedes Mischgerät über einfache Regeln "seine" Auslässe erzeugen (Bild 57b). Das Ergebnis ist bis auf zwei Ausnahmen regelmäßig und gut: Die beiden oberen Eckgeräte sind nicht vollständig mit Bereichen für die Auslässe ausgefüllt. Dieses Problem war durch Unregelmäßigkeiten in der Anordnung der Mischgeräte in (Bild 57a) vorhersehbar. Oft sind solche Unregelmäßigkeiten aber nicht vermeidbar, denn selbst ein guter



**Bild 57 a,b und c**



**Bild 58** Ein komplexes semantisches Netz (links) kann zerteilt werden, indem jeder Knoten Zentrum eines eigenen, kleinen Netzes wird (Mitte). A4 geht von der Vorstellung aus, daß jeder Knoten mit Wissen über seine Nachbarknoten ausgestattet ist und seine Nachbarknoten entsprechend beeinflussen kann. Die Abbildung rechts zeigt eine A4-Darstellung vieler kleiner semantischer Netze, von denen 11 aktiv sind und ihr Wissen auf ihre Nachbarknoten anwenden.

Entwurf ist nicht nur in seinen Randbereichen irregulär. Design ist ein Prozess, anfänglich unstrukturierte und widersprüchliche Elemente zu vervollständigen und zu ordnen. Das gelingt aber bei Problemstellungen, die nicht trivial sind, nur im Kernbereich des Designs. Deshalb sind die Unregelmäßigkeiten und Ausnahmen notwendig für den Gebäudeentwurf und ziehen das besondere Interesse eines Entwerfers auf sich.

Der algorithmische Weg aus dem Problem in Bild 57b ist das Erstellen neuer Regeln, die auch auf die Ausnahmefälle adäquat reagieren können (vgl. Bild 57c). Durch diese Erweiterung werden die Regelsätze der Mischgeräte-Container deutlich umfangreicher und komplexer, d.h. sie brauchen mehr Speicher, mehr Rechenzeit, mehr Entwicklungs- und Wartungszeit und sind schwieriger zu nutzen.

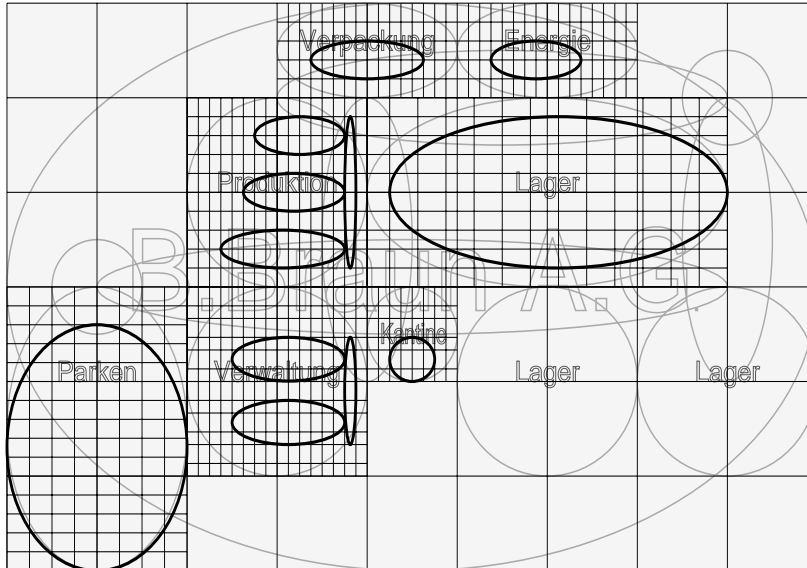
A4 verfolgt im Gegensatz zum algorithmischen Weg einen grafischen Weg: Es benutzt kleine, einfache, und schnelle Regelpakete, die nicht immer ein richtiges Ergebnis liefern, sondern in diesem Fall nur 144 der erforderlichen 148 (= 97.3%) Auslässe richtig positionieren. Das Ergebnis wird in eine kompakte grafische Darstellung projiziert. In diesem Fall konnten auf einer Fläche von 2" x 2" 184 Container und 148 Relationen so dargestellt werden, daß ein Nutzer sofort die zwei Fehler finden und mit wenigen `copy` und `paste` Kommandos beheben kann.

Als wesentliches Merkmal der Wissensmodellierung in A4 erzwingt das Modell keine Computerunterstützung. Mit den Argumenten, mit denen man auf Automatismen verzichten kann, die immer eine richtige Lösung finden (100%), kann man auch auf jede Unterstützung durch Systeme verzichten (0%) und das A4-Modell vollständig manuell betreiben. Dadurch ist es möglich, neue Bereiche des Datenraumes oder ein neues Design zunächst manuell zu erschließen (besonders flexibel, aber mit schlechter Performance), dann aber, wenn sich die Strukturen etabliert haben, schrittweise mit Automatismen zu unterstützen (weniger flexibel, aber mit höherer Performance). Die Wissensakquisition und Wissensmodellierung können sogar innerhalb des Systemes erfolgen, denn die neuen Automatismen können auf den bereits etablierten Strukturen identifiziert, entwickelt und getestet werden.

#### *Die Auswirkungen auf die Wissenskoordination*

Die leistungsfähige grafische Darstellungsmethode komplexer Strukturen im A4-Modell erlaubt, neben einer einfacheren Wissensmodellierung v.a. auf einen zentralen Koordinator (wie z.B. bei Blackboard Systemen) zu verzichten und diese Rolle einem oder mehreren Nutzern zu übertragen.

A4 bricht ein semantisches Netz auf und behandelt jeden Knoten mit seinen Nachbarknoten einzeln (Bild 58 a, b). Dadurch entstehen viele kleine und überschaubare Netze, die weitgehend unabhängig voneinander sind. Im Sinne von aktiven Objekten versucht jeder Knoten, seine unmittelbaren Nachbarn in einen



**Bild 59** Ein Übersichtsplan einer Fabrik für medizinische Geräte. Obwohl die Strukturen an die verschiedenen Orte und Anforderungen angepaßt und stark differenziert sind, hat ein Nutzer durch die besondere grafische Repräsentation einen guten Überblick über den Zustand des Gesamtsystems.

bestimmten Zusammenhang zu bringen und in diesem Zusammenhang zu halten. Dabei laufen die Programme der aktiven Objekte permanent und gleichzeitig. Die aktiven Objekte werden in A4 nicht zentral koordiniert, sondern müssen sich nur mit ihren unmittelbaren Nachbarn abstimmen. Wie die 'part-of' Relation zwischen dem Gesamtsystem und den Komponenten im Beispiel zeigt, können Hierarchien von Kontrollstrukturen aufgebaut werden. Dabei bleibt die Komplexität in einem Knoten relativ zum Gesamtnetz gering. Ein solches System konnte bisher nicht implementiert werden. Die Konzepte wurden aber geprüft (Baht 93).

Die letztendliche Kontrolle dieses Systemes sich selbst organisierender Objekte kann von den Nutzern übernommen werden, denn die grafische Darstellungsmethode von A4 ermöglicht einen sehr guten Überblick über viele dieser aktiven Objekte in verschiedenen Abstraktionsebenen (Bild 58 c).

Durch die grafische Kontrolle des Gesamtsystems ermöglicht A4, daß sich die semantischen Strukturen gut differenzieren und an spezifische Anforderungen angepaßt werden können, ohne daß ein Nutzer die Übersicht über das Gesamtsystem verliert (Bild 59).



# 7 Interfaces und das A4-Modell

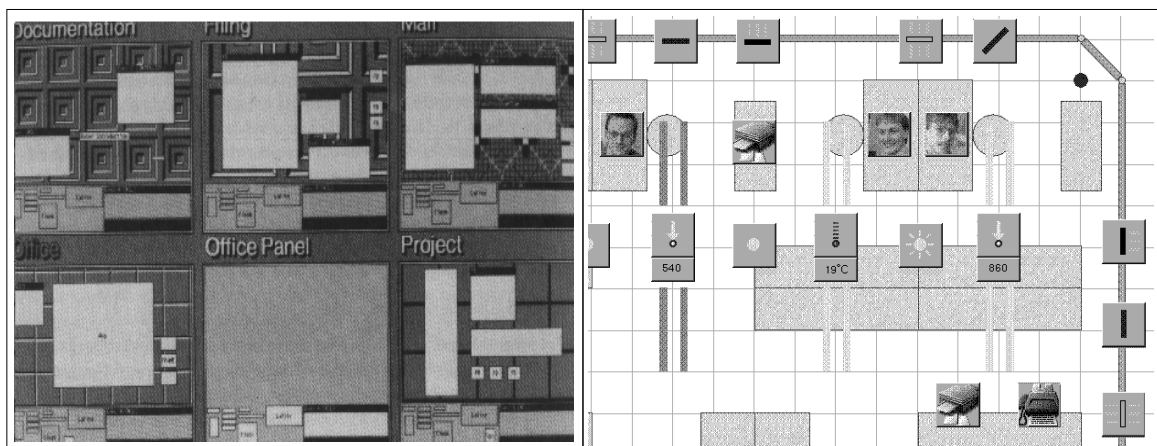
Als letzte Komponente des A4-Modells sollen die verschiedenen Aufgaben der Interfaces beschrieben werden. Man kann in komplexe, naive und allround Interfaces unterscheiden.

## komplexe Interfaces

Die bisher gezeigten Arbeitsumgebungen sind für Experten geeignet, die einen großen Teil ihrer Arbeitszeit in dem Datenraum verbringen. Das sind z.B. der Architekt, die Ingenieure oder der technische Betreuer des Gebäudes. Für sie steht nicht die leichte Verständlichkeit, sondern die Schnelligkeit und Effizienz bei komplexen Handlungen im Vordergrund. Andere Beispiele hierfür sind z.B. der EMACS Editor, der ganz auf die Unterstützung von Menüs, Grafik oder Maus verzichtend, nur mit Tastaturkürzeln das unter UNIX bevorzugteste Werkzeug für das Editieren und Debuggen von Software und Texten jeder Art ist. Konventionelle CAD Systeme mit ihren mehreren hundert Befehlen und Optionen gehören in diese Kategorie. Auch wird ein gutes Gebäudeüberwachungssystem zwar mit einem "naiven" Interface (s.u.) auf bestimmte Fehler aufmerksam machen können, aber nur mit einem "komplexen" Interface Diagnosen und Reparaturen ermöglichen.

## naive Interfaces

Für viele Nutzer von A4 sind komplexe Interfaces nicht geeignet. In der Gebäudesteuerung oder dem Facility Management sind die Nutzer z.B. mit einer großen Vielzahl spezialisierter Software konfrontiert, die sie



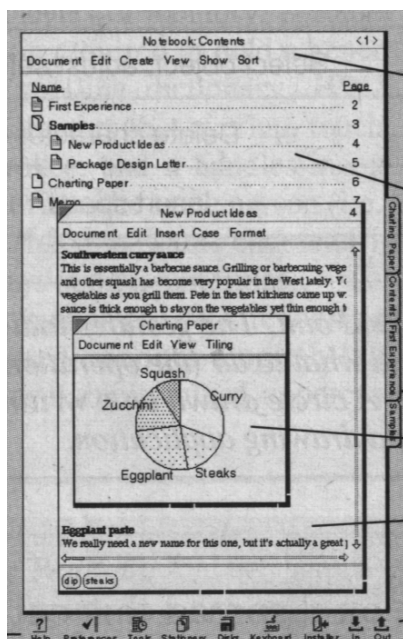
**Bild 60** Rechts: Der Übersichtsplan des Systems ROOMS (PARC), mit dem die Kommunikation innerhalb eines Bürogebäudes auf eine einfache Art organisiert werden kann. Links: Ein vergleichbares naives Interface mit A4 für Büroarbeitsplätze: Im Hintergrund ist die Möblierung von 3 Arbeitsplätzen dargestellt, im Vordergrund werden die Steuerinstrumente für die Kommunikation, Licht, Klima und Fassade angeboten.

nur selten und sporadisch nutzen müssen. Die Interfaces müssen daher so einfach und so selbstverständlich wie möglich sein (Bild 60). Ein Prototyp ist z.B. ein Leitsystem für Fluchtwege, das, abhängig von einer bestimmten Gefahr, jedem Nutzer eines Gebäudes einen sicheren Weg aus der Gefahrenzone so anzeigen muß, daß keine Mißverständnisse möglich sind. Aber auch der Informationsfluß in einem Bürogebäude oder einem Unternehmen kann sehr einfach mit einem "naiven" Interface von jedem Teilnehmer gesteuert werden.

## Allround Systeme

Ein Vorbild für A4 sind Allround Interfaces. Sie integrieren die vielfältigen alltäglichen Arbeiten in einer Art, daß sehr schnell und fast unmerklich zwischen den verschiedenen erforderlichen Programmen gewechselt werden kann. Sie integrieren Briefe, Grafiken, Ton, Telefon, Fax, Notizen, Kalender, Tabellen, Datenbank, Filestrukturen etc. in einem Informationsverbund vieler Nutzer. Beispiele hierfür sind PENPOINT (Carr 91) (Bild 61), ein modernes Betriebssystem mit Softwareumgebung für mobile Computer mit Eingabe über einen Stift, statt einer Tastatur, und HYPERCARD, das alle an einem Büroarbeitsplatz anfallenden Aufgaben ohne besondere Anforderungen integriert. Es ist vor allen Dingen besonders anpassungsfähig, weil sein Interface gezeichnet und die Programmiersprache von einem Laien gelesen und verstanden werden kann. MATHEMATICA ist ein Beispiel für den Bereich der Mathematik, mit dem nicht nur mathematische Ausdrücke symbolisch verarbeitet werden können. Es bietet gleichzeitig die Möglichkeit, Bücher zu strukturieren, zu schreiben und zu layouten, in denen die mathematischen Ausdrücke entweder ausgedruckt oder am Computer interaktiv und experimentell getestet werden können. Dabei sind Grafik, Ton, Farbe und Film integriert. Mit dieser Software verschwinden weitgehend die Grenzen zwischen Lernen, Nutzen, Experimentieren, Forschen und Entwickeln.

Da die Container im A4-Modell potentiell jedes Medium aufnehmen können, hat das A4-Modell die Anlage zu einem Allround System. Experimente auf der jetzigen Implementation mit dem Betriebssystem NeXT-Step belegen das. Doch haben die Implementation noch nicht die Konsistenz, daß es für das A4-Modell ein Allround Interface gäbe.



**Bild 61** Das Betriebssystem PENPOINT integriert Text, Grafik, Tabellen etc. in einer als Notizbuch organisierten Filestruktur für mobile Computer.

# 8 *Zusammenfassung, Ausblick*

Die Informationen, die ein Gebäudes über seine Lebenszeit betreffen, sind sehr umfangreich, heterogen und komplex. Es wurde ein Modell für ihre Integration vorgestellt: Architekten, Ingenieure und Nutzer entwickeln und betreiben ein Gebäude nicht nur in einem 3-dimensionalen sondern in einem vieldimensionalen Datenraum mit den Achsen: X, Y, Z, Zeit, Größenordnung, Auflösung, Teilsystem, Morphologie, Alternativen, User, Timetag, Komposition und Meta. Damit kann den wichtigsten Merkmalen der Informationen eines Gebäudes auf eine einheitliche Art entsprochen werden. Sie ist die Grundlage für eine einheitliche grafische Repräsentation der Informationen: Mit CAD-ähnlichen Werkzeugen können sehr viele Informationen der verschiedenen Dimensionen in kurzer Zeit in beliebigen Zusammenhängen gleichzeitig dargestellt und schnell aufeinander abgestimmt werden. Diese grafischen Darstellungs- und Editiermöglichkeiten sind Grundlage für die besondere Offenheit des Systems: neue Daten können relativ unabhängig von der Art und dem Grad ihrer Strukturierung im Datenraum angeordnet und dann schrittweise und so weit wie erwünscht an die bereits vorhandenen Strukturen angepaßt werden. Dennoch können - wenn nötig partiell - leistungsfähige Strukturen erzeugt werden, so daß die üblichen Wissensrepräsentationstechniken wie Frames, Regeln oder Constraints installiert werden können.

Das Vorhaben, alle Informationen von Gebäuden zu integrieren, ist ein großes Projekt und sprengt praktisch jeden Forschungsrahmen. Deshalb ist auch das A4-Modell nur ein Schritt in diese Richtung. Es soll in folgenden Richtungen weiter entwickelt werden:

- Mit den bisherigen Implementationen und Editoren können per Hand mindestens 50.000 Objekte im Datenraum angeordnet werden. Dabei hat das System eine üblichen CAD-Systemen entsprechende Performance. Um das System aber in der Praxis zu etablieren, wären vermutlich 1.000.000 Objekte erforderlich. Das ist nicht nur ein quantitatives, sondern auch ein strukturelles Problem. Die Grenzen der jetzigen Strukturen müßten analysiert und leistungsfähigere Strukturen entwickelt werden. Wieviele dieser qualitativen Sprünge noch nötig sind, ist nicht abzusehen.
- Die im A4-Modell entwickelte 2-dimensionale Grafiksprache muß in zwei Richtungen erweitert werden: Sie ist für einen Nutzer ohne Forschungsinteresse noch zu abstrakt und müßte deutlich mehr herkömmlichen Grafikelemente aufnehmen und sie müßte auf 3-dimensionale Darstellungstechniken übertragen werden, weil mit ihnen leistungsfähigere Interfaces zu erwarten sind.
- Das im A4-Modell benutzte Gebäudemodell ist für den praktischen Einsatz zu wenig detailliert und orientiert sich zu wenig an etablierten Klassifikations- und Arbeitsstrukturen.
- Die Integration von KI-Techniken ist noch sehr rudimentär und befindet sich noch im Experimentierstadium. Wenn es um sehr leistungsfähige Strukturen geht, kann auf sie aber nicht verzichtet werden.
- Es ist bisher auch noch nicht gelungen das A4-Modell in einer Datenbank mit einem professionellen Daten- und Transaktionsmanagement abzubilden.

# Anhang

## Anhang I: Der Forschungskontext

- Retex - Eine Integrierte Plaungs- und Verwaltungsumgebung für Klimasysteme. Planung, Simulation und Wartungsaufgaben sollen integriert werden. Dieses Projekt ist vom BMFT gefördert und wird von der Hamburger Firma Rudolf Otto Mayer geleitet (Hertkorn 90) (Hertkorn 90a), (Hertkorn 91), (Hertkorn 91b).
- XNET2 - Eine Integrierte Planungs- und Verwaltungsumgebung für Local Area Networks in Zusammenarbeit mit der Firma Digital Equipment (Raetz 89), (Ayrle 91).
- CAD/CAM Schwerpunkt - Eine interdisziplinäre Langzeitförderung verschiedener Institute der Universität Karlsruhe mit dem Schwerpunkt CAD/CAM. Das Projekt umfasst Robotik, Maschinenbau, Bauingenieurwesen, Computer-Chip layout und Architektur (seit 4.1987).
- CIM 2000 - Eine strategische Untersuchung für die Integration des Fabrik- und Produktionslayouts, in welcher Maschinenbau, Robotik und Architektur zusammenarbeiten (ab 1.92).
- OMB - Objekt-orientiertes Modellieren im Baubereich. Ein DFG Projekt in Zusammenarbeit mit einem Datenbankinstitut (Prof. Lockemann) an der Universität Karlsruhe (ab 1.92).
- Seminare und Ausbildung - in CAD (AutoCAD, GDS), Programming (AutoLISP, Common LISP (Hovestadt 89a), (Engelin 91a)).
- Symposien - 3 internationale Symposien über "Intelligent Building" (Gauchel 89), (Friedrichs 91) und "KI-Forschung im Baubereich" (Gauchel 90).
- A+ - Eine auf KI Basis entwickelte Toolbox zur Erstellung von Applikationen durch einen Nutzer um CAD Werkzeuge schnell mit spezifischer Software für spezifische Planungsaufgaben erweitern zu können (Drach et.al. 92).

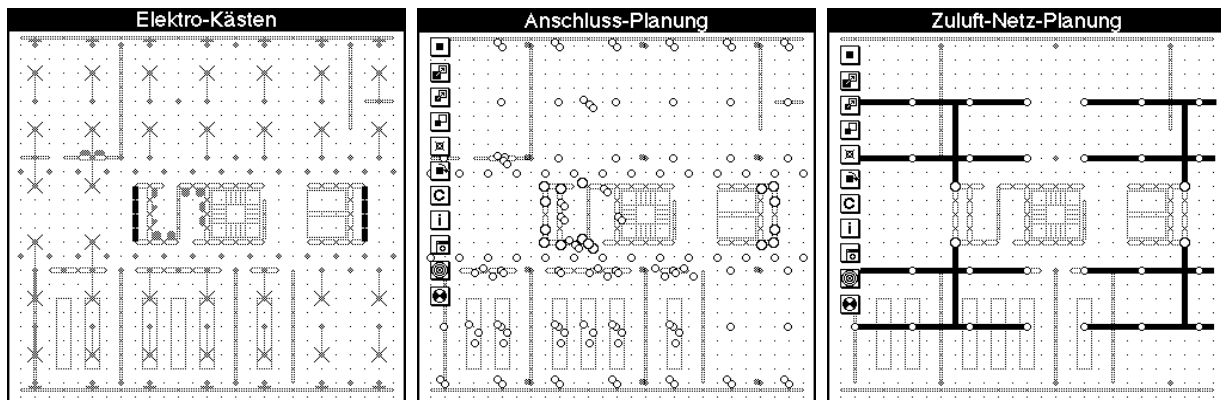
*Mit der Weiterentwicklung von A4 befasst sich das FABEL Projekt:*

- FABEL - (Fallbasierte Entwicklung wissensbasierter Systeme) Ein Verbundvorhaben des BMFT unter der Leitung der Gesellschaft für Mathematik und Datenverarbeitung in Bonn, St. Augustin mit dem Thema, fall- und regelbasiertes Wissen zu verknüpfen. A4 ist die Anwendungsdomäne dieses Projektes (FABEL 91).

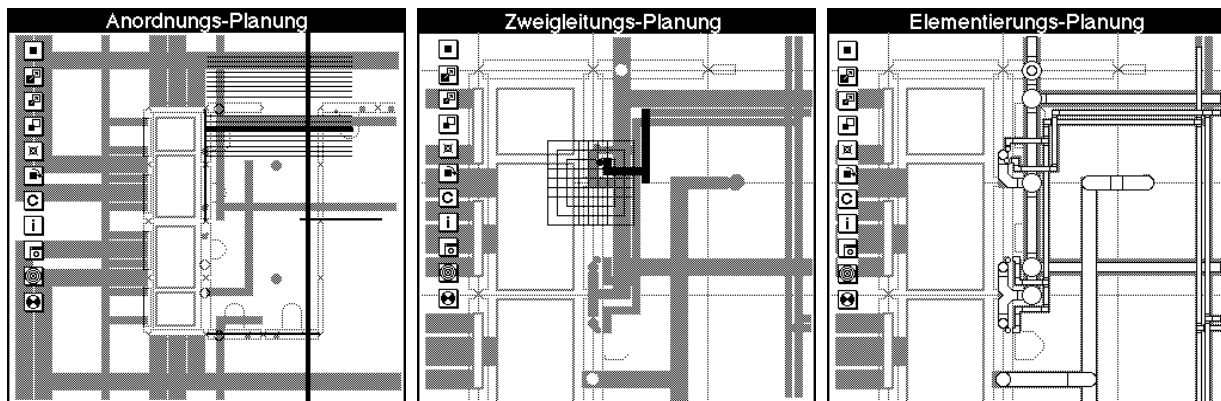
## Anhang II: armilla

Das allgemeine Installationsmodell armilla wurde am Institut für Industrielle Bauproduktion unter der Leitung von Prof. Fritz Haller 1980 - 1986 entwickelt und bietet die Möglichkeit, die technischen Systeme hochinstallierter Gebäude in gegenseitiger Abstimmung so zu planen, daß 1. definitive Planungsaussagen getroffen werden können, daß 2. die Gebäude als Baukästen realisiert und industriell gefertigt werden können und daß sie 3. besonders flexibel bleiben.

Die für diese Arbeit wichtigen Prinzipien von armilla lassen sich am einfachsten an dem Software Prototypen A2 beschreiben: Es wird unterschieden zwischen einem allgemeinen **Grundmodell** und einem spezifischen **Objektmodell**, das an ein besonderes Gebäude angepaßt wurde (Bild 62.a). Nachdem der Installationsbedarf der Endgeräte formuliert wurde (Bild 62.b), entwickelt jedes Teilsystem (hier Zuluft) ein strategisches Leitungslayout in der sogenannten **Linienleitungsplanung** (Bild 62.c). Die Planungsentscheidungen dieser Planungsstufe haben eine Unschärfe von +/- 1.20m und haben dadurch den Charakter von architektonischen Skizzen (Sie werden in A4 Zonen genannt). In der zweiten Planungsstufe, der sogenannten **Astleitungsplanung** wird jedem System konkreter Raum zugewiesen, was ohne Kollision nur in gegen-



**Bild 62** Das Objektmodell (a) und der technische Installationsbedarf des Gebäudes (b). Das strategische Linienleitungslayout der Linienleitungsplanung für die verschiedenen technischen Systeme (c) und deren räumliche Koordination in der anschließenden Astleitungsplanung (als Beispiel hier das Zuluftsystem) (d). Die Zweigleitungsplanung verbindet zwei relativ entkoppelte Planungen (e). Mit der Elementierungsplanung wird eine armilla Planung abgeschlossen (f).



seitiger Abstimmung möglich ist (Bild 62.d). Die nächste Planungsstufe wird **Zweingleitungsplanung** genannt (Bild 62.e). Hier werden die Astleitungen (Ergebnis der Astleitungsplanung) mit den konkreten Orten der Geräteanschlüsse verbunden. Die räumlich sehr komplexen Zweingleitungsgeometrien ermöglichen es, daß sowohl die Lage der Anschlüsse als auch die Lage der Astleitungen in Grenzen verschoben werden können, ohne sich gegenseitig zu beeinflussen (Diese Grenzen werden in der Linienleitungsplanung auf einem abstrakteren Planungsniveau gesetzt). Gleichzeitig ermöglicht die geometrische Entkopplung von Anschlüssen und Astleitungen, daß beide Anordnungsplanungen relativ unabhängig voneinander durchgeführt werden können, wodurch sich ihre Komplexität erheblich reduziert. Eine Planung mit armilla wird mit der **Elementierungsplanung** abgeschlossen (Bild 62.f). Hier werden innerhalb der durch die Anordnungsplanungen (Astleitungs- und Zweingleitungsplanung) reservierten Räume für die Systeme echte physikalische Bauteile angeordnet.

## Anhang III: Video<sup>1</sup>

Während in diesem Text die verschiedenen Aspekte des A4-Modells nur sehr isoliert voneinander dargestellt werden konnten, zeigt das Video das A4-Modell als Phänomen möglichst kompakt und direkt. Dabei entspricht es v.a. der Art, wie sich die entwickelte Software einem Nutzer auf dem Computerbildschirm präsentiert. Es zeigt nicht, wie die entwickelten Werkzeuge tatsächlich funktionieren, sondern wie sie sinnvollerweise funktionieren könnten. Das Rohmaterial des Videos wurde mit A4-Werkzeugen erzeugt und dann mit der Software MacroMind Director zusammengesetzt.

Das Video besteht aus drei Teilen: Eine Institutsvorstellung beschreibt den Kontext von A4, dann folgt der eigentliche Film über A4. Das Video endet mit einem Film des Prototyps A2.

## Anhang IV: Implementationen

*Das Bemühen um eine weitgehende Computerunterstützung von Entwurf, Konstruktion und Betrieb von Gebäuden hat potentiell einen dem SEQUOIA (Stonebraker 91) oder dem CYC (Lenat and Guha 89) Projekt vergleichbaren Umfang. Deshalb würden Modelle und Implementationen, die über Experimente und Prototypen hinausgingen jeden z.Zt. realisierbaren Forschungsrahmen sprengen. Interessant ist daher wie leistungsfähig die am Institut bisher entwickelten Prototypen (A1 - A4) mit welcher Implementation sind.*

---

1. Das Video ist gegen einen Unkostenbeitrag zu beziehen bei:  
Ludger Hovestadt  
Universität Karlsruhe (TH)  
Englerstraße 7  
76128 Karlsruhe  
Germany  
phone #49 721 608 2167  
fax #49 721 66 11 15  
email ludger@ifib1.ifib.uni-karlsruhe.de

## A1

A1 wurde mit Pascal und später mit CommonLisp implementiert. Es wurden Frames und Regeln zur Wissensrepräsentation benutzt. Es gab einen vorwärtsverkettenden Regelinterpreter. Auf einer PDP11 liefen Sequenzen von Planungsschritten auf verschiedenen Abstraktionsstufen ohne die Möglichkeit von Nutzereingriffen. Das System hatte eine Kapazität von ca. 50 Objekten (Mathis 88).

## A2

A2 wurde in CommonLisp und der hybriden Entwicklungsumgebung für Expertensysteme KnowledgeCraft (Frames, OPS5 und Prolog) auf microVAX und später auf SUN3 implementiert. Auf der Basis einer Blackboard Struktur entsprach ein Fachplaner (Zuluft, Abluft, Elektro ...) auf einer bestimmten Abstraktionsstufe (Bereich, Hülle, Teil) je einer knowledge source. Jede knowledge source konnte durch ein Grafikfenster dargestellt werden. Es zeigte die spezifische Sicht des Fachplaners in der jeweiligen Abstraktionsstufe auf die Daten des Blackboards. Ein Nutzer konnte die Grafiken ändern und so in der Rolle des Fachplaners die Daten des Blackboards manipulieren. Mit der knowledge source waren auch OPS5-Regeln verknüpft, die regelmäßige Nutzerinteraktionen übernehmen (Konstruktion) oder die Daten bewerten konnten (Klassifikation). Der Dialog zwischen Nutzer und den Regelsystemen über die leicht verständlichen Grafiken, die die komplexen Daten des Blackboards vermittelten, war das besondere Interesse von A2. Wichtig war auch, daß ein Nutzer frei zwischen den knowledge sources, d.h. zwischen den Fachplanern und den Abstraktionsebenen und den mit ihnen verknüpften Regelsystemen, springen konnte. Damit konnte dem iterativen Entwurfsprozess entsprochen werden. Mit diesem Blackboard System konnten ca. 250 Objekte bearbeitet und aufeinander abgestimmt werden. Es gab 12 verschiedene Regelsysteme, die jeweils verschiedene Teilaufgaben der Abwasserplanung übernehmen konnten. Die Aufgaben der anderen Fachplaner mußten manuell durchgeführt werden. A2 war ein single-user System (Drach 90), (Hovestadt 89, 90a).

## A3

A3 war der Versuch durch eine Verteilung des Blackboards und v.a. der knowledge sources auf verschiedene CommonLisp-Umgebungen die Kapazität und die Performance zu erhöhen und besonders den multi-user Betrieb zu ermöglichen. Es konnte mit multi-tasking Mechanismen des CommonLisps und mittels UNIX RPC-Protokoll verschiedene Arbeitsumgebungen gekoppelt werden. Das Projekt scheiterte jedoch, weil der Koordinationsmechanismus des Blackboard Systems mit den gleichzeitig und asynchron arbeitenden knowledge sources überfordert war. Das System wurde übermäßig komplex und für die interaktive Arbeit selbst in Experimenten zu langsam.

## A4 (UNIX - AutoCAD - KnowledgeCraft)

Mit der Einführung von Datencontainern in einem vieldimensionalen Datenraum rückte der vierte Prototyp A4 von den Blackboard Systemen ab. In der ersten Implementation von A4 entsprach ein UNIX file einem Datencontainer. Dort wurden wie in Frames Attribute und Werte des Containers in ASCII Format abgelegt. Die Achsen Auflösung, Größenordnung, Teilsystem und Morphologie des Datenraumes wurden provisorisch auf UNIX directories abgebildet. Die Achsen X, Y, Z und Zeit wurden von kleinen C-Programmen verwaltet. Durch den Gebrauch von UNIX konnten die Container von mehreren Programmen gleichzeitig bearbeitet werden. Für die Grafik gab es die Anbindung an AutoCAD, die Regeln wurden in KnowledgeCraft geschrieben. Es gab Tests mit 5 AutoCAD und 5 KnowledgeCraft Umgebungen, die gleichzeitig an denselben Containern arbeiteten. Eine Studentengruppe hat in einem Experiment an 5 AutoCAD Umgebungen gemeinsam

ein Gebäude entworfen. Mit dieser Version von A4 konnten 2000 - 3000 Container manuell bearbeitet werden. Ging man darüber hinaus, wurde v.a. das AutoLISP zu langsam. Außerdem waren die grafischen Möglichkeiten von AutoCAD unzureichend.

#### **A4 (Objective\_C, NeXTSTEP)**

Die aktuelle Version von A4 wurde in Objective\_C und NeXTSTEP geschrieben. Hier konnte die Verwaltung der Container im Datenraum in Objective\_C realisiert werden. Jetzt kann, ohne besondere Optimierung, ein Container aus 10.000 Containern in einer halben Sekunde gefunden werden. Gleichzeitig bot NeXTSTEP die besten Voraussetzungen für die Entwicklung eines leistungsfähigen und ergonomischen, grafischen Editors und zugleich auch die Integration von anderen Medien, wie Bildern, Ton, Video, Telefon etc. Die Abbildungen dieser Dissertation und auch das Rohmaterial für das Video im Anhang wurden mit diesem Editor erstellt. Mit dieser Implementation konnte problemlos (d.h. mit einer CAD-Systemen vergleichbaren Performance) ein Gebäude mit 10000 Objekten erstellt werden. Die Möglichkeiten dieser Implementation liegen mindestens bei 50000 Objekten. Das ist für ein echtes Gebäude immer noch eine Größenordnung zu wenig.

Experimente mit verschiedenen Wissensrepräsentationen (besonders modellbasiertes und fallbasiertes Wissen) auf der Basis dieser großen Menge von Objekten sind Gegenstand des aktuellen Forschungskontextes von A4 (FABEL 91). Sie werden vornehmlich in CommonLisp implementiert. Es gibt aber noch keine verwertbaren Ergebnisse.



# Glossar

## *armilla*

Das allgemeine Installationsmodell *armilla* ermöglicht die integrierte Planung der technischen Systeme --> komplexer Gebäude. Es wurde am Institut für Industrielle Baukonstruktion der Universität Karlsruhe entwickelt und ist Ausgangspunkt der Softwareversionen -->A1, A2, A3 und A4.

## *Alternative*

Eine -->Dimension. Mit ihr wird dem Phänomen entsprochen, eine komplexe Planung auf der Suche nach einer guten Lösung in mehreren Varianten zu entwickeln.

## *Auflösung*

Eine -->Dimension. Mit ihr wird dem Phänomen komplexer Planungszusammenhänge entsprochen, sich in voneinander relativ unabhängigen Planungsschichten verschiedener Genauigkeit gliedern zu lassen. A4 schlägt vor, in -->Zonen, -->Hüllen und -->Teile zu unterscheiden.

## *Ausstattung*

Eine -->Position auf der -->Morphologie-Achse. Auf ihr können -->Container angeordnet werden, die sich mit den Geräten im weitesten Sinn befassen, die für die Erfüllung der -->Nutzungen nötig sind. Dazu gehören z.B. Tische, Telephone, aber auch die Bauteile des Gebäudes.

## *A1*

Die erste Softwareversion von -->*armilla*. Mit ihr konnte gezeigt werden, dass Objekte und vorwärtsverkettennde Regeln eine geeignete Repräsentationsform für *armilla* sind.

## *A2*

Die zweite Softwareversion von -->*armilla*. Hier rückte die interaktive Steuerung der Objekte und Regeln von -->A1 in den Vordergrund. A2 ist ein Blackboard System.

## *A3*

Die dritte Softwareversion von -->*armilla*. Hier sollten die Methoden aus -->A2 auf eine multiuser-Umgebung erweitert werden. Diese Version scheiterte, weil die zentralen Steuerungen der gewählten Blackboard Architektur zu komplex wurden.

## *A4*

Die vierte Softwareversion von -->*armilla*. Hier ermöglicht das Konzept der räumlichen Anordnung von aktiven Objekten in einem vieldimensionalen -->Raum multiuser Fähigkeiten bei gleichzeitiger Nutzung von KI-Techniken.

## *A4-Modell*

Das in dieser Arbeit beschriebene Modell für eine weitgehende Computerunterstützung des Bauens.

*Bauprozeß*

Unter Bauprozeß (oder allgemein Bauen) wird in A4 die Bauplanung, -steuerung und -verwaltung eines Gebäudes während seiner Lebenszeit verstanden werden.

*Container*

Container haben die Funktionalität von Objekten objektorientierter Programmierung, haben aber zusätzlich eine -->Position im -->Datenraum und formulieren dadurch ihren Interessen- und Geltungsbereich. Außerhalb des räumlichen Bereiches eines Containers können weder seine Daten noch seine Funktionalität erfragt, manipuliert oder benutzt werden. -->Nachbarschaft

*Datenraum*

Alle Informationen in A4 werden in Form von Containern im Datenraum angeordnet. Neben den physikalischen Dimensionen x, y, z und Zeit schlägt A4 noch eine Reihe weiterer Dimensionen des Datenraumes vor: -->Dimensionen.

*Dimension*

Die Dimensionen des vieldimensionalen -->Datenraumes. Er besteht neben den physikalischen Achsen x, y, z, und Zeit je nach Komplexität des Projektes aus einer Auswahl mindestens dieser Achsen: -->Morphologie, -->Größenordnung, -->Auflösung, -->Teilsystem, -->User, -->Alternative, -->Komposition, -->Meta und -->Timetag. Die Dimensionen von -->A4 haben verschiedene Skalen, was z.B. die Begriffe der -->Nachbarschaft oder -->Position verkompliziert.

*dynamische Gebäude*

Gebäude, die konzeptionell darauf angelegt sind, sich über ihre Lebenszeit kontinuierlich ändern zu können, um sich an neue Anforderungen anpassen zu können.

*Erschließung*

Eine -->Position auf der -->Morphologieachse des -->Raumes. Auf ihr können -->Container angeordnet werden, die die -->Ausstattung eines Gebäudes technisch ermöglichen (z.B. die Stromversorgung).

*Größenordnung*

Eine diskrete und geordnete -->Dimension des -->Datenraumes. Mit ihr wird dem Phänomen komplexer Planungszusammenhänge entsprochen, die sich in voneinander relativ unabhängige Planungsschichten verschiedener Größenordnungen gliedern. In der Architektur kennt man eine Unterscheidung von Regionalplanung, Städteplanung, Gebäudeplanung und Detailplanung. Innerhalb -->komplexer Gebäude ist eine feinere Gliederung in Beschreibungen verschiedener Größenordnungen nötig und sinnvoll einzusetzen.

*Hülle*

Eine -->Position auf der -->Auflösungsachse des -->Datenraumes. Auf ihr können -->Container angeordnet werden, die räumlich soweit bestimmt sind, daß ihre räumliche Lage exakt definiert werden kann. Hüllen werden in A4 in der Regel als Reckecke dargestellt (-->Zone, -->Teil).

*integrierte Planung*

Im Gegensatz zur -->seriellen Planung eine Planungsmethode, die auf der Zusammenarbeit aller am Bauen beteiligten Personen und in interdisziplinären Planungsteams schon in frühen Planungsphasen beruht.

### *komplexe Gebäude*

werden in A4 Gebäude genannt, die die Möglichkeiten konventioneller Planungs- und Baumethoden übersteigen. Insbesondere sind dies hochinstallierte Gebäude, wie Bürogebäude, Laborgebäude, oder Schulen --> midi.

### *Komposition*

Eine Achse des -->Datenraumes. Mit ihr können -->Container auch dann zusammengefaßt werden, wenn sie auf den anderen -->Dimensionen nicht kollidieren (--> Nachbarschaft).

### *Meta*

Eine Achse des -->Datenraumes. Mit ihr können Informationen über -->Container abgelegt werden, was besonders für KI-Techniken wichtig ist.

### *Morphologie*

Eine diskrete und ungeordnete -->Dimension des -->Datenraumes. Mit ihr wird dem Phänomen verschiedener Beschreibungsformen komplexer Planungszusammenhänge entsprochen. In der Architektur kann man unterscheiden z.B. in -->Nutzungen, --> Ausstattungen, -->Erschließungen und -->Verbindungen.

### *midi*

Der Gebäudebaukasten Prof. Hallers für bis zu viergeschossige hochinstallierte Gebäude (z.B. Bürogebäude, Schulen, Laborgebäude ...). Eine Besonderheit ist die Integration auch der technischen Systeme: --> armilla.

### *Nachbarschaft*

Zwei -->Container sind benachbart, wenn sich ihre -->Positionen nur in wenigen -->Dimensionen unterscheiden. Der Begriff der Nachbarschaft wird auf den verschiedenen Achsen des -->Datenraumes unterschiedlich interpretiert. Auf den physikalischen Achsen x, y, z und t müssen sich die Container durchdringen. Auf den diskreten und geordneten Achsen wie Teilsystem oder Morphologie können sie jeden Wert annehmen. Auf den diskreten geordneten Achsen müssen sie Nachbarwerte haben. Auf der Alternativen-Achse müssen sie Nachbarposition in der Alternativenhierarchie haben. Container sind auf jeden Fall dann Nachbarn, wenn sie auf der Dimension -->Komposition die gleiche -->Position haben.

### *Nutzung*

Eine -->Position auf der -->Dimension -->Morphologie des -->Datenraumes. Auf ihr können -->Container angeordnet werden, die sich mit den Nutzungen oder Anforderungen eines Gebäudes befassen.

### *permanente Planung*

Da sich -->dynamische Gebäude kontinuierlich ändern, erfordern sie eine permanente Planung.

### *Umgebung*

Eine Umgebung eines -->Containers ist die Menge aller seiner --> Nachbarn.

### *Timetag*

Eine kontinuierliche -->Dimension des -->Datenraumes. Planungsentscheidungen werden zu einem bestimmten Zeitpunkt getroffen. Um nachverfolgen zu können, wie sich eine Planung entwickelt hat, ist es

wichtig, daß die -->Container auch auf dieser Achse angeordnet werden.

#### *Position*

Der Ort eines -->Containers im -->Datenraum.

#### *räumliche Vererbung*

Die aus der objektorientierten Programmierung bekannten Methoden der Vererbung werden in A4 auf die räumliche Hierarchie der -->Container übertragen. Wichtig dafür ist der Begriff der -->Nachbarschaft.

#### *serielle Planung*

Die zur Zeit praktizierte Planungsmethode im Bauen, die darauf beruht, daß die einzelnen beteiligten Personen mit klar abgegrenzten Kompetenzen in einer klar definierten Reihenfolge eine Planung nacheinander bearbeiten: -->integrierte Planung.

#### *Schablone*

Eine in -->armilla entwickelte Methode zur einfachen räumlichen Koordinierung von Objekten. Schablonen haben viel mit prototypischen Planungssituationen zu tun.

#### *Teil*

Eine -->Position auf der -->Auflösungs-Achse des -->Datenraumes. Auf ihr können -->Container angeordnet werden, die nicht detaillierter bearbeitet werden sollen oder wegen Zugriffsrechten nicht bearbeitet werden können. Sie werden deshalb so dargestellt, wie sie sind (z.B. als Photo): -->Zone, -->Hülle.

#### *Teilsystem*

Eine diskrete und ungeordnete -->Dimension des -->Datenraumes. Mit ihr wird dem Phänomen verschiedener einander ergänzender Systeme in komplexe Planungszusammenhänge entsprochen. In der Architektur unterscheidet man z.B. neben verschiedenen Gewerken noch Fachingenieure, die einzelne Teilgebiete, die gesamte Planung begleitend, bearbeiten.

#### *Verbindung*

Eine Sonderform der -->Erschließung für die Strecken der technischen Leitungssysteme.

#### *Zone*

Eine -->Position auf der -->Auflösungs-Achse des -->Datenraumes. Auf ihr können -->Container angeordnet werden, die räumlich relativ unbestimmt sind und die Eigenschaft von Skizzen haben. Zonen werden in A4 in der Regel in Form von Ellipsen dargestellt: --> Hülle, -->Teil.

# Literatur

- Ackoff, R.L.; 1962: *Scientific Method: Optimizing Applied Research Decisions*. 1962.
- ACS; 1991: *Architekten Computer Systeme*. 8. ACS Fachmesse mit 4. Fachkongress, Wiessbaden, 28. - 30. 11.1991.
- Agha, G.A., 1986: *ACTORS: A Model of concurrent Computations in Distributed Systems*. The MIT Press, Cambridge, MA, 1986.
- Ahmed, S.; Sriram, D.; Logcher, R.; 1990: *A Comparison of Object-Oriented Database Management Applications for Engineering Applications*. Research Report No R90-03, MIT, Cambridge, MA, 1990.
- Anderl, R.; Schilli, B.; 1987: *STEP - Eine Schnittstelle zum Austausch integrierter Modelle*. 1987.
- Anderl, R.; Grabowski, H.; Schmitt, M.; 1991: *STEP - Internationale Schnittstelle zum Produktdatenaustausch in der Rechnerintegrierten Produktion (CIM)*. Lehrgang Nr. 13806/98.121, Technische Akademie Esslingen, 1991.
- ARCH+; 1986: *Was für'n Typ?*. ARCH+, Zeitschrift für Architektur und Städtebau, 85, 6.1986.
- ARCH+; 1991: *Textile Architektur*. ARCH+, Zeitschrift für Architektur und Städtebau, 107, 3.1991.
- Ars Electronica: Interactive Arts - Jährlich stattfindende Ausstellungen, Linz, Austria.
- Asimov, M.; 1962: *Introduction to Design*. 1962.
- Atkinson, M.; Banchilhon, F.; DeWitt, D. J.; Dittrich, K.R.; Maier, D.; Zdonik, S.; 1989: *The Object-Oriented Database System Manifesto*. In: Proc. of the DOOD Conference, pages 40 -57, Kyoto, Japan, 1989.
- Augenbroe, G.; Laret, L.; 1989: *COMBINE*. Pilot Study Report CEC DGXII, 1989.
- Ayrle, H.; 1991: *Intelligent Tools for Intelligent Buildings*. In: MICAD Proceedings of the 10th Int. Conf. on CAD/CAM, Computer Graphics and Computer Aided Technologies, Volume 2, Session 15, Paris, 1991.
- Banham, P. R.; 1960: *Theory and Design in the First Machine Age*. 1960.
- Banham, P. R.; 1969: *The Architecture of the Well-tempered Environment*. 1969. Deutsche Übersetzung in: ARCH+ 93, 1988.
- Banham, P. R.; 1981: *Design by Choice*. 1981.
- Benevolo, L.; 1978: *Geschichte der Architektur des 19. und 20. Jahrhunderts*, 2 Bände, 1978.
- Bhat, R. R.; Gauchel J.; Van Wyk, S.; 1993: *Communication in Cooperative Building Design*. CAAD futures 93, 1993.
- Bijl, A.; 1989: *Computer Discipline and Design Practice - Shaping our Future*. Edinburgh, 1989.
- Caromel, D.; 1988: *A general model for concurrent and distributed OBJECT-oriented programming*. ACM SIGPLAN Workshop on OBJECT-Based Concurrent Programming, San Diego, CA, USA, 26-27 Sept. 1988, SIGPLAN Notices, vol.24, no.4, April 1989, pp. 102-104.
- Carr, R. M.; 1991: *The Point of the Pen*. BYTE 2.1991, pp. 211 - 221.

- Chamberlin, D. D.; 1990: *Managing properties in a system of cooperating editors*. IBM Research Division, 1990.
- CIB; 1991: *Computer Integrated Building*. ESPRIT - Computer Integrated Manufacturing, Exploratory Action No. 5604, 1991.
- Clarkson, M. A.; 1991: *An Easier Interface*. BYTE 2.1991, pp 277 - 282.
- Coyne, R.; 1988: *Logic Models of Design*. 1988.
- Coyne, R. F.; 1991: *ABLOOS: An Evolving Hierarchical Design Framework*. Thesis at the Carnegie-Mellon University, Pittsburgh, PA, 1991.
- David, B. T.; 1987: *Multi-Expert Systems for CAD*. In: tenHagen et.al. (eds.): *Intelligent CAD Systems I*, 1987.
- Dayal, U.; Buchmann, A. P.; McCarthy, D.R.; 1988: *Rules are Objects too: a Knowledge Model for an Active, Objectoriented Database System*. Advances in Objectoriented Database Systems. 2nd International Workshop, Eberburg, West Germany, 27-30 Sept. 1988, Springer-Verlag, vii+373, 1988, pp. 129-43
- Delattre, E.; Geib, J. M.; 1990: *OMPHALE an Active Objectbased Distributed Operating System*. Decentralized Systems. Proceedings of the IFIP WG 10.3 Working Conference, Lyon, France, 11-13 Dec. 1989, North-Holland, viii+451 pp., 1990, pp.145-58
- Descotte, Y.; Latombe, J.; 1985: *Making Compromises among Antagonist Constraints and a Planner*. AI Journal 27, pp. 183 - 217, 1985.
- Drach, A.; Gauchel, J.; Hovestadt, L.; 1990: *Ein intelligentes Planungswerkzeug für das Verlegen haustechnischer Leitungen in hochinstallierten Gebäuden*. In: VDI-Berichte Nr. 861.5, Düsseldorf, Germany, 1990.
- Drach, A.; Langenegger, M.; Heitz, S.; 1992: *Flexible Environments for Integrated Building Design*. In: Krause et.al. (eds): *CAD'92 - Neue Konzepte zur Realisierung anwendungsorientierter CAD-Systeme*, GI-Fachtagung, Berlin, 5.1992.
- Duffy, H.; 1989: *The Changing City*. London, 1989.
- Durand, J. N. L.; 1801: *Recueil et Parallele des Edifices de tout genre Anciens et Modernes*. Paris An IX, ditn "le grand Durand", 1801.
- Durand, J. N. L.; 1802: *Precis des Lecons d'Architecture dinbnees a l'Ecole Polytechnique*, Paris An X, 1802.
- Eames, C.; 1977: *Film: Powers of Ten*. 1977 (first study version 1968). Book: Morrison, P., Morrison, P., Eames, C.; Eames, R.: *Powers of Ten*. Scientific American Library, 1982.
- Eastman, Ch.; 1968: *Towards a Theory of Automated Design*. Arbeitspapier fier die Interdisciplinary Conference on Decision-aids, Ohio State University, 1968. *On the Analysis of Intuitive Design Processes*. Arbeitspapier fier die Design Methods Group Conference, 1968. *Explorations of the Cognitive Processes in Design*. ARPA Report, Dep. of Computer Science, CMU, DDC No. 671-158, 2.1968.
- Eigen, M.; 1987: *Stufen zum Leben - Die frühe Evolution im Visier der Molekularbiologie*. 1987.
- Engelin, D.; Heitz, S.; 1991: *Programmieren für Architekten in der Programmiersprache LISP*. Institut für Industrielle Bauproduktion der Universität Karlsruhe, 1991.
- Engelmore, R.; Morgan, T. (eds.); 1988: *Blackboard Systems*. 1988.
- Enkovaara, E.; Salmi, M.; Sarja, A.; 1988: *RATAS-Project - Computer Aided Design for Construction*. Helsinki, 1988.

- EXPRESS; 1989: *Exchange of Product Model Data - Part 11: The Express Language*. Document CD 10303 - 11, ISO TC184/SC4 N64, 1989.
- FABEL; 1991: *Integration von modell- und fallbasierten Entwicklungsansätzen für wissensbasierte Systeme*. Antrag für ein Verbundvorhaben, Gesellschaft für Mathematik und Datenverarbeitung, Bonn, 1991.
- Fenves, S. J.; Hendrickson, C.; Maher, M. L.; Flemming, U.; Schmitt, G.; 1990: *An Integrated Software Environment for Building Design and Construction*. In: Computer Aided Design, 1990.
- Flemming, U.; Coyne, R.; Galvin, T.; Rychener, M.; 1988: *A Generative Expert System for the Design of Building Layouts - Version 2*. In: Artificial Intelligence in Engineering: Design (Proceedings of the Third International Conference, Palo Alto, CA), J. Gero, ed., Elsevier (Computational Mechanics Publications), New York, 1988, pp 445 - 464.
- Flemming, U.; 1989: *More on the Representation and Generation of Loosely Packed Arrangements of Rectangles*. Environment and Planning B: Planning and Design 16, pp 327 - 359, 1989.
- Friedrichs, K. (ed), 1991: *2. symposium intelligen buildings*. Universität Karlsruhe, 1991.
- Gauchel, J. (ed.); 1989: *1. symposium intelligen buildings*. Universität Karlsruhe, 1989.
- Gauchel, J. (ed.); 1990: *KI-Forschung im Baubereich*. Universität Karlsruhe, 1990.
- Gauchel, J.; Hovestadt, L.; Wyk, S. Van; Bhat, R. R.; 1992: *Modular Building Models*. Design Decision Support Systems '92, Eindhoven, 1992.
- Gauchel, J.; Wyk, S. Van; Bhat, R. R.; Hovestadt, L.; 1992: *Building Modeling Based on Concepts of Autonomy*. AID '92 - Second International Conference On Artificial Intelligence In Design, Pittsburgh, 1992.
- Gero, S. J.; 1987: *Prototypes: a new schema for knowledge-based design*. Working Paper, Architectural Computing Unit, Dept. of architectural Science, Sydney, 1987.
- Gero, S. J.; Rosenman, M.; Oxman, R. E.; 1991: *What is a case*. In: Schmitt, G. (ed) : CAAD Futures '91, 1991.
- Gielingh, W. F., 1988: *General AEC Reference Model (GARM)*. TNO Building and Construction Research, BI-88-150, 1988.
- Haller, F.; 1974: *MIDI - ein offenes system für mehrgeschossige bauten mit integrierter medieninstallation*. USM baussysteme haller, Münsingen, CH, 1974.
- Haller, F.; 1975: *totale stadt, ein modell - integral urban, a global model*. Olten, CH, 1975.
- Haller, F.; Barth, A.; Zaugg, H.; Weber, H.; Scheidiger, P.; Bill, R.; 1978 - 1982: *SBB Ausbildungszentrum Löwenburg Murten, Schweiz*. Stahlbausystem MIDI.
- Haller, F.; 1983: *Planung, Betrieb und Unterhalt von Leitungsführungen in Hochinstallierten Gebäuden*. Bericht zum Forschungsvorhaben HA 1218/1-1, Institut für Industrielle Bauproduktion, Universität Karlsruhe, Germany, 1983.
- Haller, F.; 1985: *ARMILLA - ein installationsmodell*. Institut für Industrielle Bauproduktion der Universität Karlsruhe, Germany, 1985.
- Haller, F.; 1988: *bauen und forschen*. Dokumentation der Ausstellung. Solothurn, CH, 1988.
- Haller, F.; 1989: *Über die Notwendigkeit wandelbarer Gebäude*. In: Gauchel, J. (ed.): 1. symposium intelligen buildings, Universität Karlsruhe, 1989.

- Hayes-Roth, F.; Waterman, D. A.; Lenat, D.B.; (eds), 1983: *Building Expert Systems*. Addison-Wesely, 1983.
- Hayes-Roth, B.; 1984: *BB1: An Architecture for Blackboard Systems that Control, Explain, and Learn about their Behavior*. Department of Computer Science, HPP-84-16, Stanford University, CA, USA, 1984.
- Hayes-Roth, B.; Vaughan, J. M.; Garvey, A.; Hewett, M.; 1986: *Application of the BB1 Blackboard Control Architecture to Arrangement Assembly Tasks*. In: The International Journal for Artificial Intelligence in Engineering, 1 (2), pp 85 - 94, 1986.
- Hayes-Roth, B.; Garvey, M.; Vaughan, M.; 1987: *A Modular and Layered Environment for Reasoning about Action*. Knowledge Systems Laboratory, Computer Science Department, Stanford University, Reprint No. KSL 86-38, 1987.
- Hertkorn, C.; 1990: *Untersuchung verschiedener Techniken der Wissensrepräsentation in einem Expertensystem für das Verlegen haustechnischer Leitungsnetze in hochinstallierten Gebäuden*. Diplom am Institut für Industrielle Bauproduktion der Universität Karlsruhe, Germany, 1990.
- Hertkorn, C.; 1990: *Intelligent Design Environment System (IDE) - Description of the Project*. IEA 21 RN 66, 6.1990.
- Hertkorn, C.; 1990: *Requirements to the Design Support Environments - Draft*. IEA 21 RN 68/91, 6.1990.
- Hertkorn, C.; 1991: *A Distributed Expert System as an Integrated Building System*. Proceedings to the IBPSA - BS '91, 8.1991.
- HOAI; 1991: *Verordnung über die Honorare für Leistungen der Architekten und der Ingenieure in der Fassung der vierten ÄnderungsVO, Stand 1.1.1991*. Düsseldorf, 1991.
- Hoskins, E. M.; 1977: *The OXSYS System*. In Gero, J.S. (ed): *Computer Applications in Architecture*, pp 343 - 391, 1977.
- Hovestadt, L.; Haller, F.; Gauchel, J.; Drach, A.; 1989: *Ein "intelligentes CAD-System" für die Planung und Verwaltung von Leitungen in hochinstallierten Gebäuden*. In: VDI Berichte 775, Düsseldorf, Germany, 1989.
- Hovestadt, L.; Heitz, S.; 1989: *Der IBE-Lisp Kurs - Version 1.0, Einführung für Architekten*. Institut für Industrielle Bauproduktion der Universität Karlsruhe, 1989.
- Hovestadt, L.; 1990: *the armilla film*. (MacroMind Animation for Macintosh), Institute für Industrielle Bauproduktion, Karlsruhe, Germany, 1990.
- Hovestadt, L.; 1991: *A4 - A Model for An Extensive Use of Computers in Architecture*. Proceeding of the Second International Workshop on Computer Building Representation for Integration, Aix-les-Bains, 1991.
- Hovestadt, L.; 1992: *B0 - A Generative Model for an International, Integrated and Distributed Computer environment for the Building Industry*. ESPRIT III - Proposal, R&D Task IV.2.3.
- Hovestadt, V.; 1992: *Zeitraumplanung - Entwurf von Instrumenten und Darstellungstechniken für das Planen in Zeiträumen*. Diplomarbeit am Lehrstuhl für Konstruktives Entwerfen, Prof. P. Kulka, RWTH Aachen, 1992.
- IAP; 1991: *Institut für Architektur- und Planungstheorie*. Software für den Architekturbereich 1991/92, Vol.1 AVA, Vol.2 CAD, Hannover, 1991.
- ISIS Engineering Report; 1991: *Entwicklung, Konstruktion, Fertigung - CAE/CAD/CAM - PPS/ Logistik, Unternehmen, Konzepte, Programme*. Jg. 8, 1990/1991.
- Jones, J. C.; 1966: *Design Methods compared*. In: *Design*, 8-9, 1966.
- Kalay, Y. E.; Swerdloff, L. M.; Hartmann, A. C.; 1987: *A Knowledge-based Computable Model of Design*,



- Proceedings of Expert Systems in Computer Aided Design*. In: Gero, J. (ed.), 1987.
- Karagiannis, D.; (ed), 1990: *Information Systems and Artificial Intelligence: Integration Aspects*. First Workshop, Ulm, FRG, 3.1990.
- Katz, R. H.; 1985: *Information Management for Engineering Design*. Surveys in Computer Science. 1985
- KC; 1990: *Knowledge Craft 4.0*. Carnegie Group Inc. Pittsburgh, 1990.
- Keller, S.; 1986: *Baukostenplanung für Architekten*. Wiesbaden, 1986.
- Kemper, A.; Lockemann, P. C.; Moerkotte, G.; Walter, H. D.; Lang, S. M.; 1990: *Autonomy over Ubiquity: Coping with the Complexity of a distributed World*. In: Proc. Intl. Conf. Entity-Relationship Approach, Lausanne, 1990.
- Kohler, N.; 1991: *Statische Gebäude, dynamische Fassaden*. In: Friedrichs, K. (Ed.): 2. Symposium Intelligent Building, Universität Karlsruhe, Fakultät für Architektur, 10.1991.
- Kolodner, J. L.; 1992: *An Introduction to Case-Based Reasoning*. Artificial Intelligence Review, 6:3-34, 1992.
- Kuhnert, N.; 1980: *Soziale Elemente der Architektur: Typus und Typusbegriffe im Kontext der Rationalen Architektur*. Dissertation an der RWTH Aachen, 1980.
- Lenat, D. B.; Guha, R. V.; 1989: *Building Large Knowledge-Based Systems. Representation and Inference in the Cyc Project*, Addison-Wesely, 1989.
- Luckman, J.; 1970: *An Approach to the Management of Design*. In: Operation Research Quarterly, Vol. 18, pp 345-358, 1967.
- Maher, M. L.; 1987: *Engineering Design Synthesis*. AI EDAM 1(3), pp 207 - 213, 1987.
- Malfroy, S.; 1985: *Typologie als Methode der Interpretation. Der Theoretische Teil des Architekten Savario Muratori*. Werk/Bauen und Wohnen, Nr. 11, 1985.
- Malfroy, S.; 1986: *Kleines Glossar zu Muratories Stadtmorphologie*. Arch+ 85, pp 66 - 73, 1986.
- Manning, P.; 1968: *Appraisals of Building Performance and Their Use in the Design Process*. Architects Journal, 10.1968.
- Markus, T. A.; 1969: *The Role of Building Performance Measurement and Appraisal in Design Method*. In: Broadbent, G., Ward, A. (eds.): Design Methods in Architecture, 1969.
- Mathis, Ch.; 1988: *Installationsplanung mit einem wissensbasierten System*. Dissertation am Institut für Industrielle Bauproduktion der Universität Karlsruhe, 1988.
- Mattos, N.; 1990: *An Approach to DBS-Based Knowledge Management*. In: Karagiannis, D. (ed): Information Systems and Artificial Intelligence: Integration Aspects. First Workshop, Ulm, FRG, 3.1990.
- McLuhan, M.; 1964: *Understanding Media*, 1964.
- Minsky, M.; 1975: *A framework for representing knowledge*. In: Winston ed.: The psychology of computer vision. McGraw Hill, 1975
- Mitchel, T.; Steinberg, L.; Shulman, J.; 1985: *A knowledge-based approach to design*. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985.
- Morrison, P.; Morrison, P.; 1982: *Powers of Ten*. Scientific American Library, 1982.
- Müller, H.; 1989: *Die Gebäudehülle von morgen*. In: Gauchel, J. (ed.): KI-Forschung im Baubereich. Uni-

versität Karlsruhe, 1989.

Müller, J.; Franz, L.; 1989: *Anforderungen des Konstrukteurs an die Entwicklung von Wissenssystemen für das Konstruieren*. In: VDI Berichte 775 , pp 129 - 172, 1989

Muratori, S.; 1959: *Studi per una operante Storia urbana di Venezia*. Istituto Poligrafico dello Stato, Rom, 1959. Deutsche Übersetzungen und Lehrgänge: ETH Zürich, Lehrstuhl für Geschichte des Städtebaus.

Negroponte, N.; 1970: *The Architecture Machine*. MIT Press, MA, USA, 1970.

Negroponte, N.; 1975: *Computer Aids to Design and Architecture*. 1975.

Neufert, E.; 1943: *Bauordnungslehre. Handbuch für Rationelles Bauen nach Geregelmtem Maß*. 1.Auflage 1943, wird ständig neu aufgelegt und aktualisiert.

Newell, A.; Simon, H. A.; 1972: *Human Problem solving*. New York. 1972

Nijssen, H.; 1989: *Conceptual Schema and Relational Database Design*. Prentice Hall, 1989.

ORBIT1; 1982: *The ORBIT 1 Study* (DEWG/EOSYS) . London, 1983.

ORBIT2; 1985: *The ORBIT 2 Study* (DEWG/EOSYS). London, 1985.

Oxman, R. E.; 1991: *Adapting Prior Design: A Case-Based Reasoning Approach to Knowledge-Based Design*. Proceeding to the Second International Workshop on Computer Building Representation for Integration, Aix-les-Bains, France, 1991.

Pahl, G.; Beitz, W.; 1977: *Konstruktionslehre*. 1977.

Panerai, P.; 1980: *Typologien*. Arch+ 50, pp 7 - 16, 1980.

Pawley, M.; 1990: *Theory and Design in the Second Machine Age*. 1990.

Payne, F. W.; (ed.), 1989: *Strategies for Energy Efficient Plants & Intelligent Buildings*. 1989.

Piel, R.; 1978: *Ordnen, Suchen, Finden - Bauinformationen mit dem BRD/SfB-System*. Köln, 1978.

Pohl, J.; Reys, I.; 1988: *An Integrated Intelligent CAD Environment*. 4th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany, 8.1988.

Powers, B. R.; 1980: *The Global Village*. 1980

Puppe, F.; 1988: *Einführung in Expertensysteme*. Studienreihe Informatik. Berlin, 1988.

Puppe, F.; 1990: *Problemlösungsmethoden in Expertensystemen*. Studienreihe Informatik. Berlin, 1990.

Raetz, P.; 1989: *XNET: Ein intelligentes CAD-System für die Planung von Lokalen Netzwerken in Gebäuden*. Dissertation am Institut für Industrielle Bauproduktion der Universität Karlsruhe, Germany. Fortschritt-Berichte VDI, Reihe 20: Rechnerinterstützte Verfahren, Nr. 19, Düsseldorf, 1989.

Rao, K. V. B.; Gafni, A.; Raeder, G.; 1989: *Dynamo: a model for a distributed multi-media information processing environment*. Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences. Vol.II: Software Track (IEEE Cat. No.89TH0243-6), Kailua-Kona, HI, USA, 3-6 Jan. 1989, Washington, DC, USA, IEEE Comput. Soc. Press, xvii+1106 pp., 1989, pp. 800-9 vol.2

Rehak, D. R.; Howard, H. C.; Sriram, D.; 1984: *Architecture of an Integrated Knowledge Based Environment for Structural Engineering Applications*. In: Gero, J.S. (ed): *Knowledge Engineering in Computer Aided Design*, 1984.

- Richter, P.; 1988: *Entwicklung einer integrierten Informationsstruktur für relationale Datenbanken im Bauwesen*. Dissertation an der Gesamthochschule Kassel, 1988.
- Rittel, H.; 1970: *Der Planungsprozess als iterativer Vorgang von Varietätserzeugung und Varietätseinschränkung*. In: Entwurfsmethoden in der Bauplanung (Joedicke, J. (ed)), Arbeitsberichte zur Planungsmethodik 4, 1970.
- Rosenman, M. A.; Gero, J. S.; Oxman, R. E.; 1991: *What's in a case: the use of case bases, knowledge bases and databases in design*. In: Schmitt, G. (ed): CAAD futures '91, Zürich, 1991.
- Roth, K., 1982: *Konstruieren mit Konstruktionskatalogen*. 1982.
- Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, R.; Lorenzen, W.; 1991: *Object-Oriented Modeling and Design*, 1991.
- Saiedian, H.; Unger, E. A.; 1990: *ABSL: an actor-based specification language for office automation, Cooperation*. 1990 ACM Eighteenth Annual Computer Science Conference Proceedings, Washington, DC, USA, 20-22 Feb. 1990, New York, NY, USA, ACM, xxv+475 pp., 1990, pp. 252-8
- Sapaty, A.; 1988: *WAVE-1: A New Ideology of Parallel and Distributed Processing on Graphs and Networks*. In: Future Generations Computer Systems 4, pp 1 - 14, 1988.
- Sapaty, A.; 1990: *The WAVE Model for Advanced Knowledge Processing*. Alexander von Humboldt Fellowship Programme, FRG and Dept. of Engineering Science, Univ. of Oxford, 1991.
- Schank, R. C.; Abelson, R.P.; 1977: *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. 1977.
- Schank, R. C.; 1982: *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, 1982.
- Shih S. G.; 1991: *Case-based Representation and Adaptation in Design*. in: Schmitt, G. (ed): CAAD futures '91, Zürich, 1991.
- Simon, H. A.; 1973: *The structure of ill-structured problems*. Artificial Intelligence Journal, vol 4, pp 181-201, 1973.
- Smithers, T.; 1992: *Design as Exploration: Puzzle-Making and Puzzle-Solving*. Position statement at the AID92, EDRG, CMU, Pittsburgh, 6.92.
- Sowa, J.F.; 1984: *Conceptual Structures*. 1984.
- Stefik, M.; 1981: *Planning with Constraints, MOLGEN Part 1, Planning and Meta-Planning, MOLGEN Part 2*. AI-Journal 16, 111 - 169, 1981.
- STEP-2DBS; 1989: *DIN-NAM 96.4.2., STEP-2D Building Subset, version 1.0, Standardized exchange format*. RIB e.V., Stuttgart, 1989.
- Stonebraker, M.; Dozier, J.; 1991: *Large Capacity Object Servers to Support Global Change Research*. Technical Report Nr. 91/1, Electronic Research Laboratory, University of California, Berkeley, CA, 7.91.
- Stratton, R. C.; Jarrell, D. B.; 1990: *Towards the Development of Multilevel-Multiagent Diagnostic Aids*. Chapter four : Intelligent Diagnostic Systems, London, 1990.
- Suter, P.; Geller, R.; Kohler, N.; Gilst, J. van; 1986: *Haustechnik in der Integralen Planung*. Impulsprogramm Haustechnik 1986, Bundesamt für Konjunkturfragen, Bern, Schweiz, 1986.
- Takala, T.; 1987: *Intelligence beyond Expert Systems: A Physiological Model with Applications in Design*. In: Intelligent CAD Systems I (ten Hagen et. al. (Eds.)), 1987.

- Tomiyama, T.; ten Hagen, P. J. W.; 1987: *Representing Knowledge in Two Distinct Descriptions: Extensional vs. Intensional*. CWI Report No. CS/R8728, Centre for Mathematics and Computer Science, Amsterdam, 1987.
- TRON; 1989: *Tron-Concept Intelligent House*. Architect K. Sakamura, Tokyo 1986 - 1989.
- Veth, B.; 1987: *An Integrated Data Description Language for Coding Design Knowledge*. Intelligent CAD Systems I (ten Hagen et. al. (Eds.)), 1987.
- Waard, M. de; Tolman, F.; 1991: *A Model to Describe Residential Buildings*. Second international Workshop on Computer Building Representation for Integration, Aix-les-Brains, France, 1991.
- Wielinga, B.; Schreiber, G.; Breuker, J.; 1992: *KADS: A modelling approach to knowledge engineering. Knowledge Aquisition*. Special Issue on KADS, 1992.
- Wike, P. R.; Kennicott, P. R.; 1989: *ISO STEP Baseline Requirements Document (IPIM)*. ISI TC 184/SC4/WG1 N 284, ISO Draft Proposal No. 10103, 1989.
- Worthington, J.; 1989: *Building Intelligence - a global Perspective*. In: Gauchel, J. (ed.): 1. symposium intelligent buildings, Universität Karlsruhe, 1989.
- Worthington, J.; 1991: *Intelligent Workplaces*. In: Friedrichs, K. (ed.): 2. Symposium Intelligent Buildings. Universität Karlsruhe, 1991.